AD-A013 733

EVALUATION OF PROBABILITY OF DETECTION FOR SEVERAL TARGET FLUCTUATION MODELS

David A. Shnidman

Massachusetts Institute of Technology

Prepared for:

Electronic Systems Division

8 July 1975

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-75-109	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Evaluation of Probability of Detection for Several Target Fluctuation Models		5. TYPE OF REPORT & PERIOD COVERED Technical Note
		6. PERFORMING ORG. REPORT NUMBER Technical Note 1975-35
7. AUTHOR(s) Shnidman, David A.		8. CONTRACT OR GRANT NUMBER(s) F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 2027 Program Element No. 63208F
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Systems Command, USAF Andrews AFB Washington, DC 20331		12. REPORT DATE 8 July 1975
		13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

envelope detection
radar target detection
Marcum model
Swerling models
log-normal model
probability of detection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Highly efficient algorithms have been derived for calculating the probability of detection, P_D, for several target fluctuation models. These include the Marcum model (constant target cross section) and the four Swerling models (chi-squared fluctuations). Programs are presented in computer free notation as well as FORTRAN (for accuracies of 10^{-6} and 10^{-12}) for each of the models.

In addition, an approximate expression for the elevation of P_D, applicable when the target return is constant during a scan but exhibits log-normal fluctuation from scan-to-scan, is presented, together with a listing of a FORTRAN program that evaluates the expression. An efficient iterative routine for the determination of the threshold from the probability of false alarm is also included.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD13733

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

EVALUATION OF PROBABILITY OF DETECTION FOR SEVERAL TARGET FLUCTUATION MODELS

D. A. SHNIDMAN

Group 44

TECHNICAL NOTE 1975-35

8 JULY 1975

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

ABSTRACT

Highly efficient algorithms have been derived for calculating the probability of detection, P_D, for several target fluctuation models. These include the Marcum model (constant target cross section) and the four Swerling models (chi-squared fluctuations). Programs are presented in computer free notation as well as FORTRAN (for accuracies of 10^{-6} and 10^{-12}) for each of the models.

In addition, an approximate expression for the elevation of P_D, applicable when the target return is constant during a scan but exhibits log-normal fluctuation from scan-to-scan, is presented, together with a listing of a FORTRAN program that evaluates the expression.

An efficient iterative routine for the determination of the threshold from the probability of false alarm is also included.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TABLES

SECTION 1

INTRODUCTION

Some highly efficient algorithms for the evaluation of probability of detection on N incoherently integrated returns have been determined and are presented here. The efficiency is, of course, inversely related to accuracy, consequently two FORTRAN versions are given: one with accuracy of 10^{-6}, and the other with an accuracy of 10^{-12}.

In April 1960, the IRE Transactions on Information Theory published papers by Marcum and Swerling [Refs. 1, 2, 3] that derived expressions for the probability of detection, P_D, for targets of constant backscattering (Marcum) and of four different models for backscatter fluctuation (Swerling). The resulting formulas were manipulated into more convenient forms [Refs. 4, 5, 6,], and graphs supplementing those of Marcum and Swerling were furnished. Heidbreder and Mitchell [Ref. 7] added the log-normal model for characterizing the fluctuation. This latter model has been found to be very useful for targets that have large mean to median ratios. Other models exist [Refs. 8, 9, 10], but we will limit ourselves to these since they cover most situations well enough.

Although the graphs provided in the references are useful, an efficient means of computation is sometimes necessary. Two problems arise if one uses the expression as presented. First, computations can be quite lengthy, especially if great accuracy is required, and second, there can be problems of overflow and underflow even though all answers are between zero and one. New efficient expressions have been obtained for the Marcum and Swerling models, and these are modified to reduce the computation effort further without sacrificing accuracy. An approximation is presented for the case in which the radar cross section is constant within a scan but exhibits log-normal fluctuations from scan to scan.

Programs for the evaluation of the various $P_D(s)$ are presented in computer free language and in two FORTRAN versions for an IBM 370; the first with an accuracy of 10^{-6} and the second with an accuracy of 10^{-12}.

SECTION 2

EXPRESSIONS FOR PROBABILITY OF DETECTION

In this section the expressions for P_D for the six cases are presented without proof since they are derived in the literature [Refs. 1, 2, 3, 4, 5, 6, 7]. In order to preserve the Swerling model numbers 1 through 4, we will refer to the constant target as case 0, the Swerling cases as cases 1 through 4, and the log-normal fluctuation as case 5.

2.1 Case 0:

The Marcum derived expression for P_D is

$$_0P_N(\overline{X}, Y) = \int_Y^{\infty} \left(\frac{v}{\overline{X}_N}\right)^{\frac{N-1}{2}} e^{-(v + \overline{X}_N)} I_{N-1}\left(2\sqrt{\overline{X}_N v}\right) dv \tag{2.1}$$

where N is the number of pulses incoherently integrated

Y is the threshold level

$\overline{X}$ is the average signal-to-noise ratio of a single pulse

$\overline{X}_N$ is the total signal-to-noise ratio of all N pulses

$I_n(Z)$ is the nth order modified Bessel function.

If the N pulses are not identical, we still have

$$\overline{X} = \overline{X}_N / N \quad .$$

By substituting in Eq. (2.1) the infinite series for $I_n(Z)$

$$I_n(Z) = \left(\frac{Z}{2}\right)^n \sum_{k=0}^{\infty} \frac{(Z/2)^{2k}}{k!(n+k)!} \tag{2.2}$$

and interchanging the order of summation and integration, one obtains Fehlner's equation [Ref. 4]

$$_0P_N(\overline{X}, Y) = e^{-\overline{X}_N} \sum_{k=0}^{\infty} \frac{(\overline{X}_N)^k}{k!} \sum_{m=0}^{N-1+k} e^{-Y} \frac{Y^m}{m!} \tag{2.3}$$

2.2 Case 1:

The signal-to-noise ratio, X, with mean $\overline{X}$, is assumed constant throughout the integration scan but distributed as chi squared with 2 degrees of freedom from scan to scan. The resulting P_D is [Refs. 5, 6]

$$_1P_N(\overline{X}, Y) = \begin{cases} e^{-\frac{Y}{1+\overline{X}}} & \text{for } N = 1 \\ 1 - \displaystyle\sum_{m=N-1}^{\infty} e^{-Y} \frac{Y^m}{m!} & \text{for } N \geq 2 \end{cases}$$

$$+\left(1 + \frac{1}{\overline{X}_N}\right)^{N-1} \sum_{m=N-1}^{\infty} e^{-\left(\frac{Y}{1+\frac{1}{\overline{X}_N}}\right)} \frac{\left(\frac{Y}{1+\frac{1}{\overline{X}_N}}\right)^m}{m!} \tag{2.4}$$

2.3 Case 2

The signal-to-noise ratio X, with mean $\overline{X}$, is assumed to be distributed as chi-squared with 2 degrees of freedom and independent from pulse to pulse. The expression for P_D is [Refs. 5, 6]

$$
{}_2P_N(\overline{X}, Y) = 1 - \sum_{m=N}^{\infty} e^{-\frac{Y}{1+\overline{X}}} \frac{\left(\frac{Y}{1+\overline{X}}\right)^m}{m!} \quad . \tag{2.5}
$$

2.4 Case 3

The signal-to-noise ratio, X, with mean $\overline{X}$, is assumed to be distributed as chi-squared with 4 degrees of freedom from scan to scan but constant within a scan. The expression for P_D is [Refs. 5, 6]

$$
{}_3P_N(\overline{X}, Y) = \begin{cases} e^{-\frac{Y}{1+\overline{X}/2}} \left(1 + \frac{(\overline{X}/2)\,Y}{(1+\overline{X}/2)^2}\right) & \text{for } N = 1 \\[2ex] \frac{Y^{N-1} e^{-Y}}{(N-2)!} \frac{1}{(1+\overline{X}_N/2)} + \sum_{m=0}^{N-2} e^{-Y} \frac{Y^m}{m!} & \text{for } N \geq 2 \end{cases}
$$

$$
+ e^{-\frac{Y}{1+\overline{X}_N/2}} \left(1 + \frac{1}{\overline{X}_N/2}\right)^{N-2} \left[1 - \frac{N-2}{\overline{X}_N/2} + \frac{Y}{1+\overline{X}_N/2}\right]
$$

$$
\times \left[1 - \sum_{m=0}^{N-2} e^{-\frac{Y}{1+\overline{X}_N/2}} \frac{\left(\frac{Y}{1+\overline{X}_N/2}\right)^m}{m!}\right] \quad . \tag{2.6}
$$

2.5 Case 4:

The signal-to-noise ratio, X, with mean $\overline{X}$, is assumed to be distributed as chi-squared with 4 degrees of freedom and independent from pulse to pulse. The expression for P_D is [Refs. 5, 6]

$$_4P_N(\overline{X}, Y) = 1 - \left(\frac{1}{1 + \overline{X}/2}\right) \sum_{k=0}^{N} \frac{N!}{k!(N-k)!} \left(\frac{\overline{X}}{2}\right)^k \sum_{m=N+k}^{\infty} e^{-\frac{Y}{1 + \overline{X}/2}} \frac{(1 + \overline{X}/2)^m}{m!} \tag{2.7}$$

2.6 Case 5:

The signal-to-noise ratio, X, with mean $\overline{X}$, is assumed to be log-normally distributed [Ref. 7] from scan to scan but constant within a scan. The log-normal distribution with parameters $\overline{X}$ and ρ is

$$p_X(x|\overline{X}, \rho) = \frac{1}{\sqrt{2\pi}\,\sigma x} e^{-\frac{\ln^2\left(\frac{x}{M}\right)}{2\sigma^2}} \qquad 0 \le x < \infty \tag{2.8}$$

where $M = \frac{\overline{X}}{\rho}$ is the median of X and $\sigma = \sqrt{2 \ln \rho}$ is the variance of $\ln X$. The expression for P_D is

$$_5P_N(\overline{X}, Y, \rho) = \int_0^{\infty} {}_0P_N(x, Y)\, p_X(x|\overline{X}, \rho)\, dx \tag{2.9}$$

SECTION 3

NEW EXPRESSIONS FOR PROBABILITY OF DETECTION

In this section the cxpressions for cases 0 through 4 are modified so as to achieve more efficiently a specified accuracy and also to eliminate the problem of overflow and underflow. The efficiency is a result of the 3 facts: (1) a recursive method is used to evaluate the probability, (2) the bound on the error consists of two factors, one in terms of $\overline{X}_N$ and one in terms of Y, each of which is approaching zero, and (3) the terms used for the bound are similar to those used in the evaluation and require little additional calculation.

3.1 Case 0:

We begin with Eq. (2.3) and change the order of summation (see Fig. 1) to obtain

$$ {}_0P_N(\overline{X}, Y) = e^{-(\overline{X}_N + Y)} \left[\sum_{m=0}^{N-1} \frac{Y^m}{m!} \sum_{k=0}^{\infty} \frac{\overline{X}_N^k}{k!} + \sum_{m=N}^{\infty} \frac{Y^m}{m!} \sum_{k=m+1-N}^{\infty} \frac{\overline{X}_N^k}{k!} \right] $$

$$ = \sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} + \sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N^k}{k!} \right) \qquad (3.1) $$

since

$$ \sum_{k=0}^{\infty} \frac{X^k}{k!} = e^X \quad . \qquad (3.2) $$

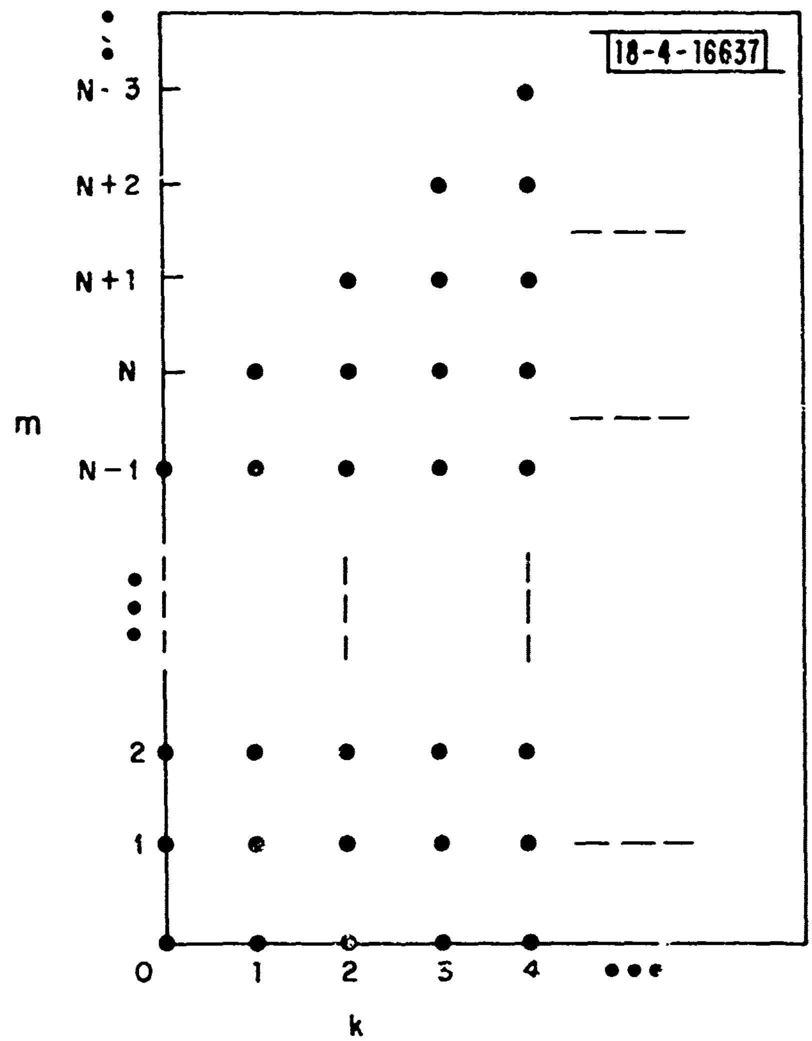

Fig. 1. (k, m) pairs contributing to the sum ${}_0P_N(\bar{X}, Y)$.

We separate Eq. (3. 1) into two terms, $P_L(\overline{X}, Y)$, where with $L \geq N$

$$P_L(\overline{X}, Y) = \sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} + \sum_{m=N}^{L} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N{}^k}{k!}\right) \tag{3. 3}$$

and

$$R_L(\overline{X}, Y) = \sum_{m=L+1}^{\infty} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N{}^k}{k!}\right) \tag{3. 4}$$

so that, if we use $P_L(\overline{X}, Y)$ to represent ${}_0P_N(\overline{X}, Y)$, our truncation error will be $R_L(\overline{X}, Y)$. We can bound $R_L(\overline{X}, Y)$ by

$$R_L(\overline{X}, Y) \leq e^{-Y} \sum_{m=L+1}^{\infty} \frac{Y^m}{m!} \left(1 - e^{-\overline{X}_N} \sum_{k=0}^{L+1-N} \frac{\overline{X}_N{}^k}{k!}\right)$$

$$= \left(1 - e^{-Y} \sum_{m=0}^{L} \frac{Y^m}{m!}\right) \left(1 - e^{-\overline{X}_N} \sum_{k=0}^{L+1-N} \frac{\overline{X}_N{}^k}{k!}\right). \tag{3. 5}$$

If we choose L large enough such that

$$\left(1 - e^{-Y} \sum_{m=0}^{L} \frac{Y^m}{m!}\right) \left(1 - e^{-\overline{X}_N} \sum_{k=0}^{L+1-N} \frac{\overline{X}_N^{\,k}}{k!}\right) \le \epsilon \tag{3.6}$$

where ϵ is our desired accuracy, then we have guaranteed that

$${}_0P_N(\overline{X}, Y) - P_L(\overline{X}, Y) \le \epsilon . \tag{3.7}$$

A program in computer-free notation to evaluate $P_L(\overline{X}, Y)$ is shown in Fig. 2. The notation used is based on one devised by Iverson [Ref. 11].*

---

*The arrow notation within the box symbolizes a specification, that is, the statement SUM $\leftarrow$ YMS is translated to mean that the quantity SUM is specified by YMS. A branch is denoted by an arrow outside the box leading to the next statement to be executed. A comparison is denoted by a colon (:), and a branch is executed if the comparison condition specified on the arrow is satisfied; otherwise the next instruction in sequence is executed. As an example, we interpret the segment below as follows: if M = N-1, go to line n+5, otherwise set M to M+1. Multiply YM by Y, divide by M, and set YM to this quantity. Set YMS to YMS + YM, set SUM to YMS, and branch to line n.

Line number	
	⋮
n	M : N - 1 (=)
n + 1	M $\leftarrow$ M + 1
n + 2	YM $\leftarrow$ YM · Y/M
n + 3	YMS $\leftarrow$ YMS + YM
n + 4	SUM $\leftarrow$ YMS
n + 5	XKS $\leftarrow$ 1. 0
	⋮

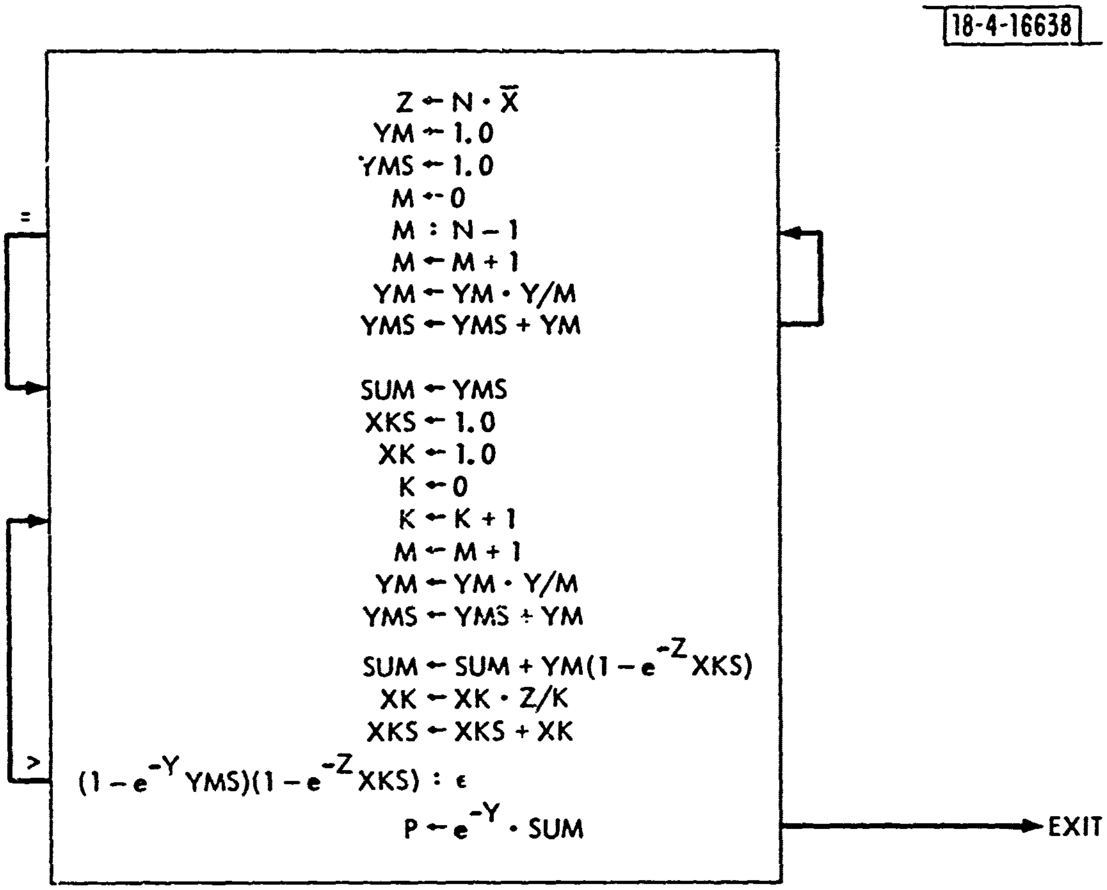

Fig. 2. Program for evaluating $P_L(\bar{X}, Y)$.

The program, as it stands, suffers from two drawbacks. First, there is a large region of the $\overline{X}_N$ - Y plane in which ${}_0P_N(\overline{X}, Y)$ is equal to one to within ϵ. We can therefore avoid, without loss of accuracy, many wasted computations by determining approximately this region in the $\overline{X}_N$ - Y plane and by testing for the given N to see if $(\overline{X}_N, Y)$ is in it. This is done by noting that for each ϵ we can determine a value $K_\epsilon(X)$ (see Appendix B) such that if

$$K < K_\epsilon(X) \tag{3.8}$$

then

$$\sum_{p=0}^{K} e^{-X} \frac{X^p}{p!} \leq \epsilon/2 \quad . \tag{3.9}$$

Similarly we can find value $M_\epsilon(Y)$ such that if

$$N > M_\epsilon(Y) \tag{3.10}$$

then

$$\sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} \geq 1 - \epsilon/2 \quad . \tag{3.11}$$

It follows that if both Eq. (3.8) and $N + K > M_\epsilon(Y)$ are satisfied then

$$
{}_0P_N(\overline{X}, Y) = \sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} + \sum_{m=N}^{N+K} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N{}^k}{k!}\right)
$$

$$
+ \sum_{m=N+K+1}^{\infty} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N{}^k}{k!}\right)
$$

$$
\geq \sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} + \sum_{m=N}^{N+K} e^{-Y} \frac{Y^m}{m!} (1 - \epsilon/2)
$$

$$
+ \sum_{m=N+K+1}^{\infty} e^{-Y} \frac{Y^m}{m!} \left(1 - \sum_{k=0}^{m-N} e^{-\overline{X}_N} \frac{\overline{X}_N{}^k}{k!}\right)
$$

$$
\geq \sum_{m=0}^{N+K} e^{-Y} \frac{Y^m}{m!} - \frac{\epsilon}{2} \sum_{m=N}^{N+K} e^{-Y} \frac{Y^m}{m!}
$$

$$
\geq 1 - \epsilon/2 - \epsilon/2
$$

$$
= 1 - \epsilon \tag{3.12}
$$

so to within an accuracy ϵ, ${}_0P_N$ is equal to one. Second, problems of overflow and underflow still exist. Underflow can be treated by determining if X (or Y) is larger than a quanti' . E where e^{-E} is at (or near) the smallest number representable by the computer. If X is greater than E, so that evaluating e^{-X} would cause an underflow, we rewrite the term

$$e^{-X} \frac{X^k}{k!} = e^{-\left[X - k\ln X + \sum_{n=1}^{k} \ln n\right]} \equiv e^{-\mathrm{Exp}} \qquad (3.13)$$

and compare Exp to E. If Exp is greater than E, then certainly $e^{-\mathrm{Exp}} << \epsilon$ and can be ignored. We then increase k until Exp < E and start our summation from that value of the index. The Y terms can be similarly handled.

If Y is small enough relative to N, the sum

$$\sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!}$$

by itself may be within ϵ of 1, and this is tested in the program.

If Y is large relative to X, then ${}_0P_N(\overline{X}, Y)$ is small and, depending on N, may be within ϵ of zero. A test is made in the program (see Fig. 4, KD:X) to determine if setting ${}_0P_N$ to zero is appropriate.

We have assumed that the accuracy of the exponential function is greater than ϵ and its errors can be ignored. It is necessary, however, to consider round-off errors. Since we are working in floating point, relative errors are small for a multiplication: and since all our answers lie between zero and one, the absolute error is less than relative error and can be ignored.

For addition, round-off errors can be important. If the algorithm of Fig. 2 terminates, then the answer is accurate to within ϵ. Round-off error, however, can prevent it from terminating. The problem is exemplified by the following situation: suppose $\overline{X}_N$ is zero,

$$\sum_{m=0}^{L-1} e^{-Y} \frac{Y^m}{m!} < 1 - \epsilon$$

so that the criterion, Eq. (3. 6), is not met, and

$$e^{-Y} \frac{Y^L}{L!} < \delta$$

where

$$\delta = \left(\sum_{m=0}^{L-1} e^{-Y} \frac{Y^m}{m!} \right) 10^{-n}$$

and n is the number of significant digits. Then, because of round-off error,

$$\left. \sum_{m=0}^{L} e^{-Y} \frac{Y^m}{m!} \right|_{\substack{\text{as represented} \\ \text{by the computer}}} = \left. \sum_{m=0}^{L-1} e^{-Y} \frac{Y^m}{m!} \right|_{\substack{\text{as represented} \\ \text{by the computer}}}$$

so that the algorithm will not end since the criterion, Eq. (3.6), will never be met. The larger the value of Y, the smaller each individual term in the sum and the greater the number of significant terms. For any ϵ and any finite number of significant digits, there exists a large enough Y such that our criterion will never be met.

We therefore have a trade-off between the number of significant digits carried, the accuracy, and the maximum values for $\overline{X}$ and Y. If $Z = \max(\overline{X}, Y)$, the worst case situation is $N = 1$ and $\overline{X} = Y = Z$. We consider some specific cases. With $\epsilon = 10^{-6}$ and double precision (16.8 digits, or more precisely 56 bits), the upper limit on Z was not found: it is greater than 1 million. With $\epsilon = 10^{-12}$ and double precision, an upper limit between 110,000 and 115,000 exists on Z. For single precision (7.2 significant digits or 24 bits), Fig. 3 shows approximate upper limits on Z for the different ϵ. If one can be satisfied with the limits single precision imposes on Z for a given ϵ, then computational efficiency will be derived from using it; otherwise double precision must be used.

The final program is shown in Fig. 4, and the double precision FORTRAN version with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$ is given in Appendix A.

Two final points that should be mentioned are (1) since it will be useful in computing some of the latter $P_D(s)$, the value of $e^{-Y}\ Y^{N-1}/(N-1)!$ is also an output of our program for ${}_0P_N(\overline{X}, Y)$. For values of Y for which ${}_0P_N(\overline{X}, Y)$ is set to 1 or 0, we choose $e^{-Y}\ Y^{N-1}/(N-1)! = 0$, and (2) the relationship between ${}_0P_N(\overline{X}, Y)$ and the Q - function, which is defined by

$$Q_N(\alpha, \beta) = \int_{\beta}^{\infty} v\left(\frac{v}{\alpha}\right)^{N-1} e^{-\frac{\alpha^2 + v^2}{2}} I_{N-1}(\alpha v)\, dv \qquad (3.14)$$

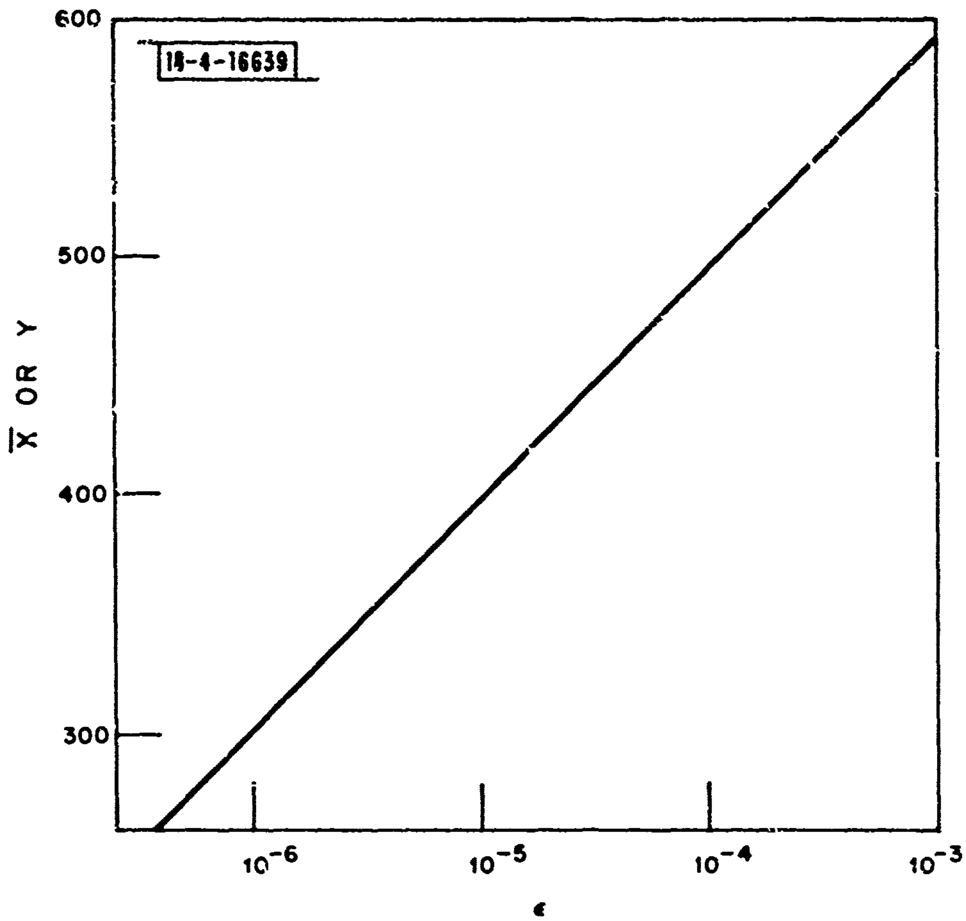

Fig. 3. Approximate limits on $\overline{X}$ and Y vs ϵ, for single precision.

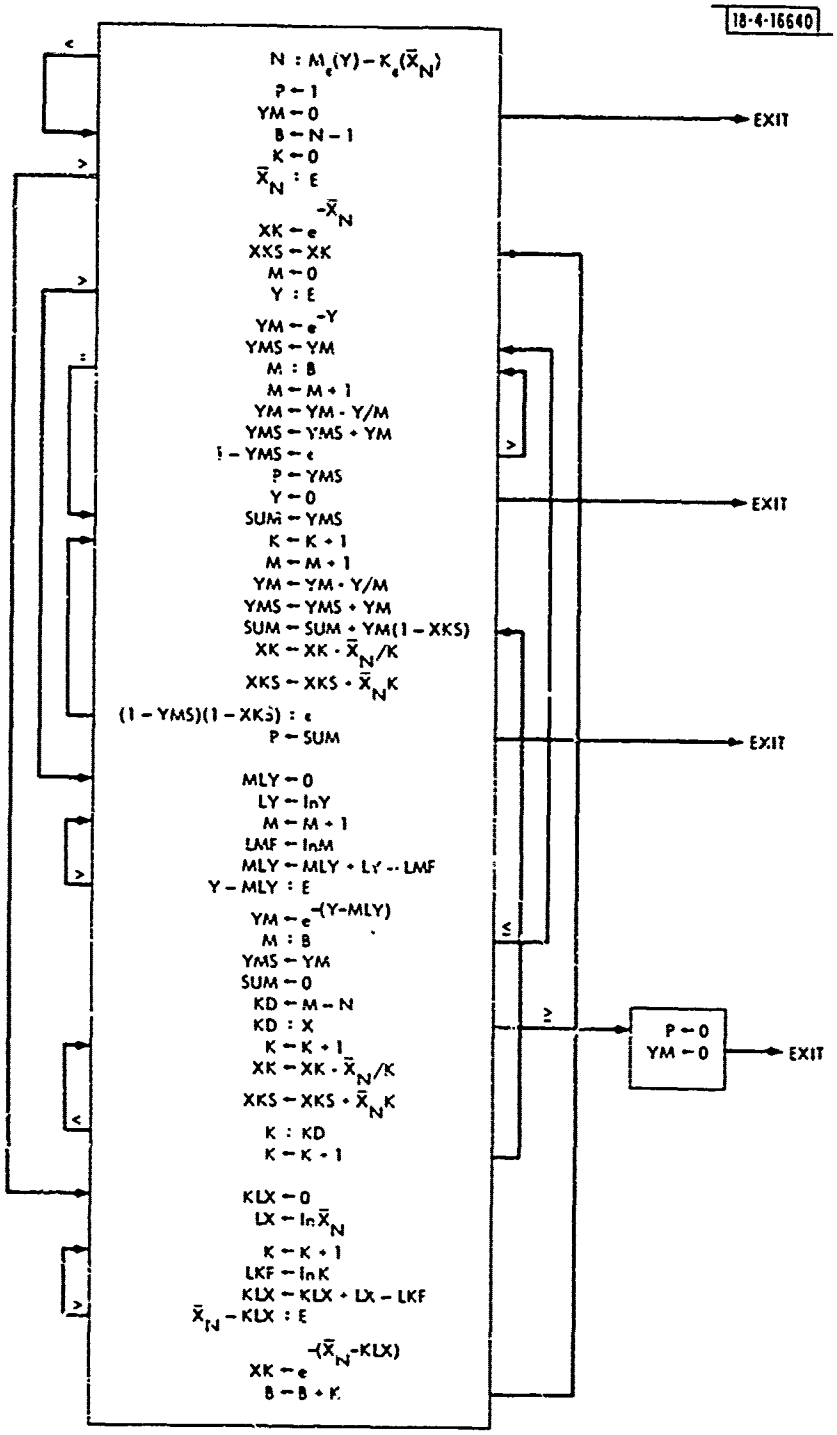

Fig. 4. Program for the Marcum Model, Case 0.

is simply

$$ {}_0P_N(\overline{X}, Y) = Q_N\left(\sqrt{2\overline{X}_N}, \quad \sqrt{2Y}\right) . \tag{3.15} $$

3.2 Case 1:

Now that we have a highly efficient means of calculating ${}_0P_N(\overline{X}, Y)$, we can use ${}_0P_N(0, Y)$ in evaluating ${}_1P_N(\overline{X}, Y)$. In terms of ${}_0P_N(0, Y)$, Eq. (2.4) becomes

$$ {}_1P_N(\overline{X}, Y) = \begin{cases} e^{-\frac{Y}{1+\overline{X}}} & \text{for } N = 1 \\ \\ {}_0P_{N-1}(0, Y) & \text{for } N \geq 2 \end{cases} \tag{3.16} $$

$$ +\left(1 + \frac{1}{\overline{X}_N}\right)^{N-1} \left[1 - {}_0P_{N-1}\left(0, \frac{Y}{1+\frac{1}{\overline{X}_N}}\right)\right] e^{-\frac{Y}{1+\overline{X}_N}} . $$

Since the term in the square bracket is bounded between zero and one, then if

$$ \left(1 + \frac{1}{\overline{X}_N}\right)^{N-1} e^{-\frac{Y}{1+\overline{X}_N}} < \epsilon/10 \tag{3.17} $$

we need not calculate the second term for $N \geq 2$. The program for ${}_1P_N(\overline{X}, Y)$ is shown in Fig. 5, and FORTRAN versions for $\epsilon = 10^{-6}$ and 10^{-12} are given in Appendix A.

3.3 Case 2:

In terms of ${}_0P_N(\overline{X}, Y)$, ${}_2P_N(\overline{X}, Y)$ becomes simply

$$ {}_2P_N(\overline{X}, Y) = {}_0P_N \left(0, \frac{Y}{1 + \overline{X}}\right) \quad . \tag{3.18} $$

3.4 Case 3:

Rewriting Eq. (2.6) in terms of ${}_0P_N(\overline{X}, Y)$ we obtain

$$ {}_3P_N(\overline{X}, Y) = \begin{cases} e^{-\frac{Y}{1 + \overline{X}/2}} \left(1 + \dfrac{(\overline{X}/2)\, Y}{(1 + \overline{X}/2)^2}\right) & \text{for } N = 1 \\[2ex] \dfrac{Y^{N-1} e^{-Y}}{(N-2)!} \dfrac{1}{(1 + \overline{X}_N/2)} + {}_0P_{N-1}(0, Y) & \text{for } N \geq 2 \end{cases} $$

$$ + \left(1 + \frac{1}{\overline{X}_N/2}\right)^{N-2} e^{-\frac{Y}{1 + \overline{X}_N/2}} \left[1 - \frac{N-2}{\overline{X}_N/2} + \frac{Y}{1 + \overline{X}_N/2}\right] $$

$$ \times \left[1 - {}_0P_{N-1}\left(0, \frac{Y}{1 + \overline{X}_N/2}\right)\right] \quad . \tag{3.19} $$

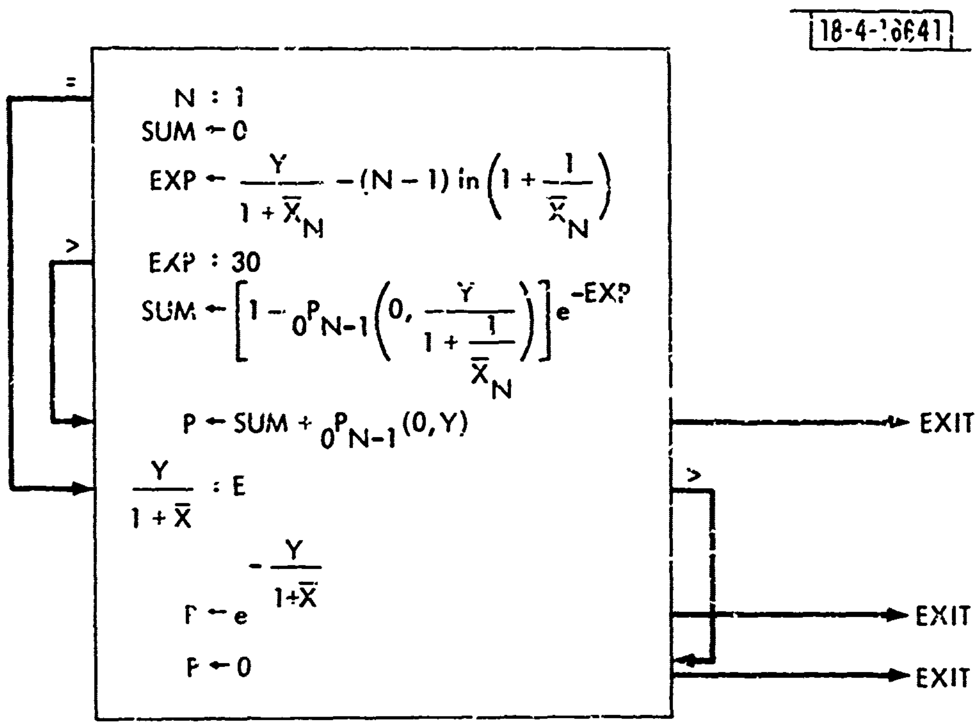

Fig. 5. Program for Swerling, Case 1.

since $0 \le 1 - {}_0P_N\left(0, \frac{Y}{1 + \bar{X}_N/2}\right) \le 1$, we first check to see if

$$e^{-\left[\frac{Y}{1 + \bar{X}_N/2} - (N - 2)\, \ell n \left(1 + \frac{1}{\bar{X}_N/2}\right)\right]} \left[1 - \frac{N - 2}{\bar{X}_N/2} + \frac{Y}{1 + \bar{X}_N/2}\right] < \epsilon \quad . \tag{3.20}$$

in which case the last term for $N \ge 2$ can be ignored. As mentioned above, we save the term $e^{-Y}\ Y^{(N-2)}/(N-2)!$ in calculating ${}_0P_{N-1}(0, Y)$, and this is used for the first term. The program is shown in Fig. 6, and FORTRAN versions for $\epsilon = 10^{-6}$ and 10^{-12} are given in Appendix A.

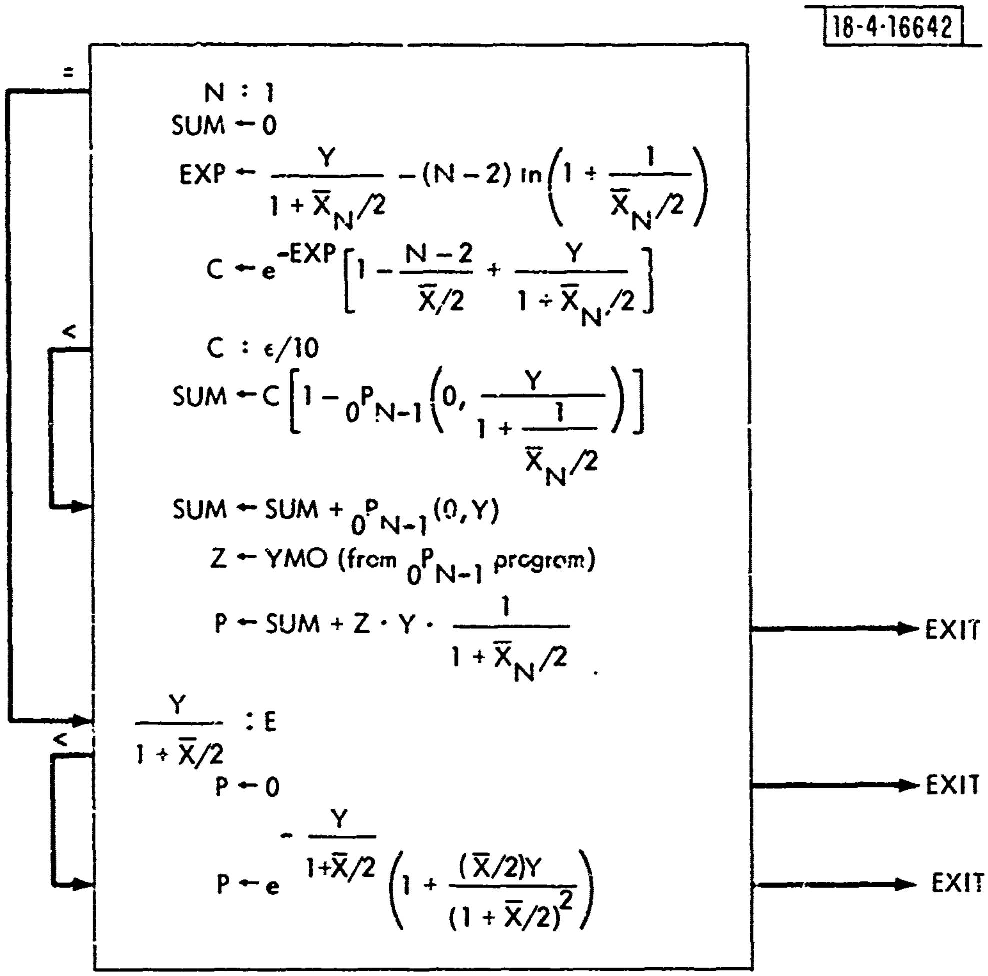

Fig. 6. Program for Swerling, Case 3.

3.5 Case 4:

Equation (2.7) is manipulated by interchanging the order of summation to obtain

$$
\begin{aligned}
{}_4P_N(\overline{X}, Y) = 1 - \left(\frac{1}{1+\overline{X}/2}\right)^N \Bigg[& \sum_{m=N}^{2N-1} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \sum_{k=0}^{m-N} \frac{N!\left(\frac{\overline{X}}{2}\right)^k}{k!\,(N-k)!} \\
& + \sum_{m=2N}^{\infty} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \sum_{k=0}^{N} \frac{N!\left(\frac{\overline{X}}{2}\right)^k}{k!\,(N-k)!} \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
= 1 - \left(\frac{1}{1+\overline{X}/2}\right)^N \Bigg[& \sum_{m=N}^{2N-1} e^{-\frac{1}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \sum_{k=0}^{N} \frac{N!\left(\frac{\overline{X}}{2}\right)^k}{k!\,(N-k)!} \\
& + \left(1+\frac{\overline{X}}{2}\right)^N \sum_{m=2N}^{\infty} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \Bigg]
\end{aligned}
$$

$$
= \sum_{m=0}^{2N-1} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} - \sum_{m=N}^{2N-1} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \left[\sum_{k=0}^{m-N} \frac{N!}{k!(N-k)!} \left(\frac{\overline{X}/2}{1+\overline{X}/2}\right)^k \left(\frac{1}{1+\overline{X}/2}\right)^{N-k}\right]
$$

$$
= {}_0P_N\left(0, \frac{Y}{1+\overline{X}/2}\right) + \sum_{m=N}^{2N-1} e^{-\frac{Y}{1+\overline{X}/2}} \frac{\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} \left[1 - \sum_{k=0}^{m-N} \frac{N!}{k!\,(N-k)!} \left(\frac{\frac{\overline{X}}{2}}{1+\overline{X}/2}\right)^k \left(\frac{1}{1+\overline{X}/2}\right)^{N-k}\right]
$$

(3.21)

There are two cases in which we can simplify the calculation of ${}_4P_N(\overline{X}, Y)$. The first is the case where $Y/(1 + \overline{X}/2)$ is small enough, relative to N, that ${}_0P_N\left(0, \frac{Y}{1 + \overline{X}/2}\right)$ is within ϵ of 1, and we set ${}_4P_N(\overline{X}, Y)$ to 1 ignoring the second term. The second case is when $\frac{Y}{1+\overline{X}/2}$ is so large, relative to N, that

$$\sum_{m=0}^{2N-1} \frac{e^{-\frac{Y}{1+\overline{X}/2}}\left(\frac{Y}{1+\overline{X}/2}\right)^m}{m!} < \epsilon$$

and we can set ${}_4P_N(\overline{X}, Y)$ to zero. This translates into a test of the form

$$2N - 1 < K_\epsilon\left(\frac{Y}{1 + \overline{X}/2}\right) \tag{3.22}$$

where $K_\epsilon\left(\frac{Y}{1 + \overline{X}/2}\right)$ the same function as in Eq. (3.8) and depends on ϵ. The program for ${}_4P_N(\overline{X}, Y)$ is shown in Fig. 7, and FORTRAN versions for $\epsilon = 10^{-6}$ and 10^{-12} are given in Appendix A.

18-4-16643

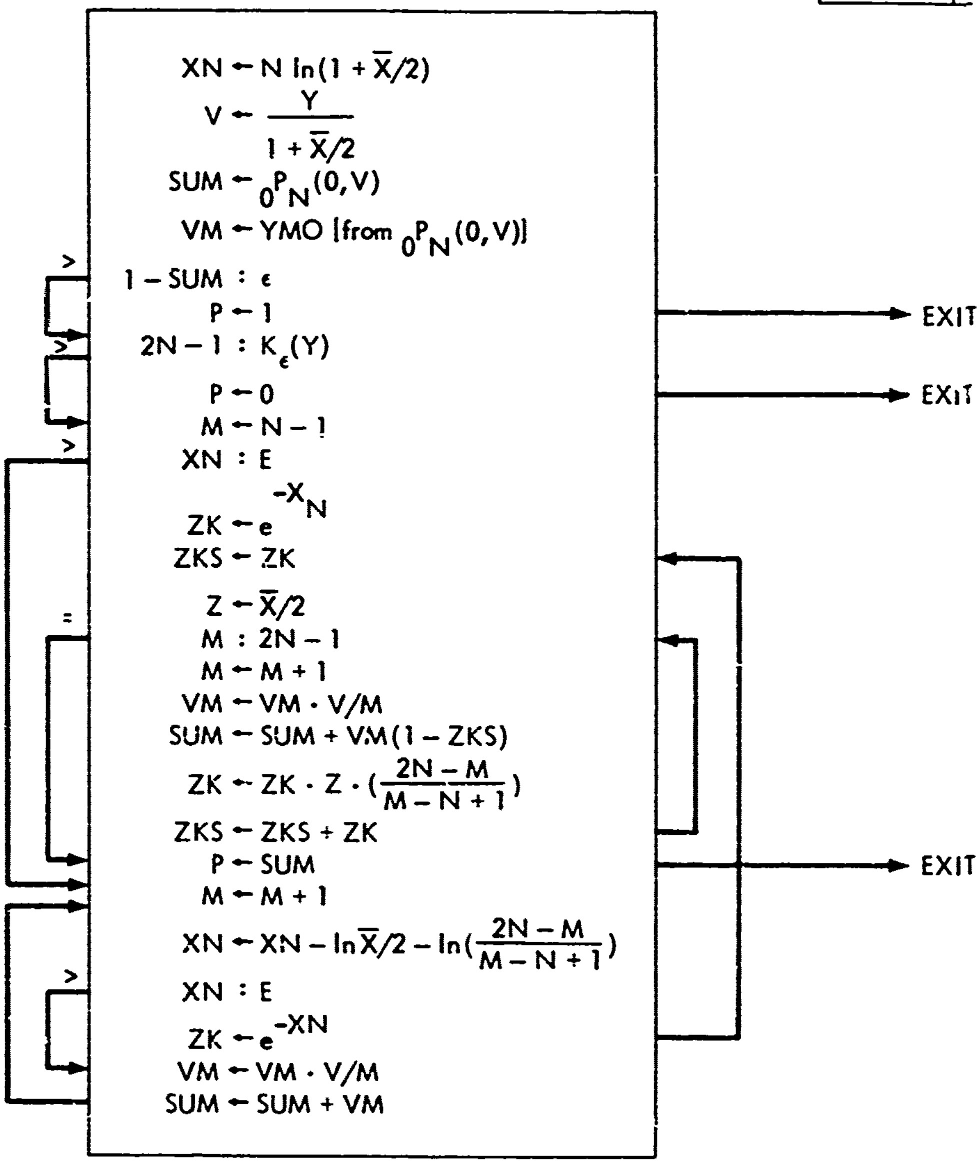

Fig. 7. Program for Swerling, Case 4.

SECTION 4

APPROXIMATION FOR CASE 5, LOG-NORMAL FLUCTUATION

In this section an approximation is presented for ${}_5P_N(\overline{X}, Y, \rho)$. Kramer et al., [Ref. 12] suggested in their Appendix B that for N = 1, ${}_5P_N(X, Y, \rho)$ can be approximated by

$$
{}_5P_1(\overline{X}, Y, \rho) \approx \frac{1}{2}\,\mathrm{erfc}\left(\frac{\ln\left(\frac{Y}{M}\right)}{\sqrt{2}\sigma}\right) \tag{4.1}
$$

where

$$
\mathrm{erfc}\,(X) \equiv 1 - \frac{2}{\sqrt{\pi}} \int_0^X e^{-t^2}\,dt
$$

is the complementary error function. No mention of accuracy is given, and it became apparent that it is dependent mainly on the parameters Y, X, and ρ, the accuracy deteriorating if Y is large or if $\overline{X}/\rho^3$ is near Y.

We will generalize the result for any N and consider the accuracy problem. One way of looking at the approximation is to consider that we have replaced ${}_0P_N(\overline{X}, Y)$ by zero for $\overline{X}$ less than some X_1, and by one for $\overline{X}$ greater then X_1. Then,

$$ {}_5P_N(\overline{X}, Y, \rho) \approx \int_{X_1}^{\infty} p_X(x | \overline{X}, \rho)\, dx \equiv I(X_1) \quad . \tag{4. 2} $$

If we define $I_1(X_1)$ and $I_2(X_1)$ as

$$ I_1(X_1) = \int_0^{X_1} {}_0P_N(x, Y)\, p_X(x | \overline{X}, \rho)\, dx \tag{4. 3} $$

and

$$ I_2(X_1) = \int_{X_1}^{\infty} [1 - {}_0P_N(x, Y)]\, p_X(x | \overline{X}, \rho)\, dx \tag{4. 4} $$

then our error, E_5, is

$$ E_5 = I_1(X_1) - I_2(X_1) \quad . \tag{4.5} $$

${}_0P_N(X, Y)$ is monotonic increasing in X from zero to one and, since we are approximating this function by zero for $X < X_1$ and by one for $X < X_1$, we want

$$ {}_0P_N(X_1, Y) = \frac{1}{2} \quad . \tag{4.6}$$

NX_1 would then be the median of the distribution density

$$ {}_0P_N(x \mid Y) = \sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} e^{-x} \frac{x^{m-N}}{(m-N)!} \quad 0 \le x < \infty \tag{4.7}$$

but would also be very cumbersome to determine. Instead, we use the mean of Eq. (4.7) for NX_1

$$ N X_1 = \sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} \int_0^{\infty} e^{-x} \frac{x^{m-N+1}}{(m-N)!} \, dx $$

$$ = e^{-Y} Y^N \sum_{m=0}^{\infty} \frac{Y^m (m+1)!}{(N+m)! \; m!} $$

$$ = e^{-Y} \sum_{m=N}^{\infty} \frac{(m+1-N) Y^m}{m!} $$

$$ = e^{-Y} Y \sum_{m=N}^{\infty} \frac{Y^{m-1}}{(m-1)!} - (N-1) e^{-Y} \sum_{m=N}^{\infty} \frac{Y^m}{m!} $$

$$= \begin{cases} Y & \text{for } N = 1 \\ Y - (N-1) + e^{-Y} \left[(N-1)\, Y \sum_{m=0}^{N-2} \frac{Y^m}{m!} + \frac{Y^{(N-1)}}{(N-2)!} \right] & \text{for } N \geq 2 \end{cases}$$

$$= \begin{cases} Y & \text{for } N = 1 \\ [Y - (N-1)]\,[1 - P_{FA}] + e^{-Y} \frac{Y^N}{(N-1)!} & \text{for } N \geq 2 \end{cases} \tag{4.8}$$

where P_{FA}, the probability of false alarm, is defined by Equation (5.1). When P_{FA} is very small, which is usually the case, we can ignore it as well as the term

$$e^{-Y} \frac{Y^N}{(N-1)!}$$

and obtain as our value for X_1

$$X_1 = \frac{Y - (N-1)}{N} \quad . \tag{4.9}$$

NX_1 is near the median and it is shown in Appendix C that ${}_0P_N\left(\frac{Y-(N-1)}{N}, Y\right)$ approaches 1/2 as Y approaches infinity; i.e., as $Y \to \infty$, NX_1 becomes the median. Using Eq. (4.9), the generalization of Eq. (4.1) becomes

$$
{}_5P_N(\overline{X}, Y, \rho) \approx I\left(\frac{Y-(N-1)}{N}\right) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\ln \frac{Y-(N-1)}{NM}}{\sqrt{2}\sigma}\right). \tag{4.10}
$$

Since our approximation

$$
{}_0P_N(X, Y) = \begin{cases} 0 \text{ for } & X < \frac{Y-(N-1)}{N} \\ 1 \text{ for } & X > \frac{Y-(N-1)}{N} \end{cases} \tag{4.11}
$$

improves rapidly with increasing N, then our approximation for ${}_5P_N(\overline{X}, Y, \rho)$ improves rapidly with increasing N. The worst case is, therefore, for $N = 1$.

We now consider the error in this approximation. We have

$$
{}_5P_N(\overline{X}, Y, \rho) = I(X_1) + I_1(X_1) - I_2(X_1) \tag{4.12}
$$

and, with X_1 chosen as in Eq. (4.9), I_1 and I_2 are not only small for most parameter values but also about equal and tend to cancel out each other. For most cases we have

$$
\left| I_1\left(\frac{Y-(N-1)}{N}\right) - I_2\left(\frac{Y-(N-1)}{N}\right) \right| < 0.005 \quad . \tag{4.13}
$$

For small Y (< 10) and/or large N, I_1 and I_2 are very small; but for large Y and small N an accuracy of at least .005 relies on I_1 and I_2 cancelling.

Since the functions ${}_0P_N(X, Y)$ in I_1 and $[1 - {}_0P_N(X, y)]$ in I_2 are reasonably symmetric about $\frac{Y - (N - 1)}{N}$ we will have good cancellation provided that $p_X(X_1 - \delta|\overline{X}, \rho)$ does not differ too greatly from $p_X(X_1 + \delta|\overline{X}, \rho)$ for small δ. $p_X(x, \rho)$ reaches a peak at $x = \overline{X}/\rho^3$ of

$$p_X(\overline{X}/\rho^3|\overline{X}, \rho) = \frac{\rho^2}{2\sqrt{\pi \ln \rho}\ \overline{X}} \tag{4.14}$$

and if this peak is high, it must be a spiked function since the area under $p_X(x|\overline{X}, \rho)$ is unity. Therefore, if the peak is large and $\overline{X}|\rho^3$ near X_1, then I_1 and I_2 are not likely to cancel. This combination of circumstances, however, is usually of little interest because the peak in Eq. (4.14) is large only if ρ is very near 1 or larger than 15. In the first case this corresponds to a nearly constant cross section, and case zero is more appropriate than case 5, and for the latter, $\overline{X}/\rho^3$ is small and is near X_1 only for very large P_{FA}. In that case, the terms that were dropped in Eq. (4.8) are no longer negligible, and Eq. (4.9) can not reasonably be used for specifying X_1.

SECTION 5

THRESHOLD LEVEL

The threshold level, Y, is related to the probability of false alarm, P_{FA}, by

$$P_{FA} = e^{-Y} \sum_{m=0}^{N-1} \frac{Y^m}{m!} = {}_0P_N(0, Y) \qquad (5.1)$$

where N is the number of incoherent integrations. DeLong and Hofstetter [Ref. 13] suggest solving for Y in Eq. (5.1) for a given P_{FA} by solving for the root of f(Y) where

$$f(Y) = \ln \frac{{}_0P_N(0, Y)}{P_{FA}} \quad . \qquad (5.2)$$

Employing the Newton-Raphson technique [Ref. 14] and noting that

$$f'(Y) = -\frac{1}{{}_0P_N(0, Y)} \frac{Y^{N-1}}{(N-1)!} e^{-Y} \qquad (5.3)$$

then

$$Y_{K+1} = Y_K + \frac{\ln \dfrac{{}_0P_N(0, Y_K)}{P_{FA}}}{\dfrac{e^{-Y_K} \, Y_K^{\,N-1}/(N-1)!}{{}_0P_N(0, Y_K)}} \tag{5.4}$$

where, since $|f'(Y)| < 1$, Y_k converges to the root of f(Y). A FORTRAN coding (see Appendix A) using ${}_0P_N(X, Y)$, in which starting values for Y are determined by

$$Y_0 = \begin{cases} N^{1.15} - 2 \log_{10} P_{FA} & N < 40 \\ \\ N - 8 \log_{10} P_{FA} & N \geq 40 \end{cases} \tag{5.5}$$

The underflow-overflow technique used in the program for case 0 (Eq. (3.13)) is employed here also. The convergence was very rapid, usually requiring only three or four iterations before the Y_k and Y_{k+1} differed by less than 10^{-12}. For very large N, a relative error of 10^{-12} can be maintained, but the absolute may exceed 10^{-12} due to round-off errors. For a small absolute error requirement and large N, the Y_k(s) can oscillate between two values of Y instead of converging.

SECTION 6

GENERALIZED FORMULATION

The results of Cases 0 to 4 have been generalized and the notation unified by Swerling [Refs. 15, 16] by the introduction of parameter K. If the chi-square distribution is generalized to a gamma distribution with parameters K and $\overline{X}$

$$p_X(\xi \mid K, \overline{X}) = \frac{\xi^{K-1}}{\Gamma(K)} \left(\frac{K}{\overline{X}}\right)^K e^{-K\xi/\overline{X}} \quad 0 \le \xi < \infty$$

then

$0 < K < 1$	corresponds to	Weinstock Case [Ref. 9]
$K = 1$	corresponds to	Case 1
$K = 2$	corresponds to	Case 3
$K = N$	corresponds to	Case 2
$K = 2N$	corresponds to	Case 4
$K = \infty$	corresponds to	Case 0

The probability of detection [Ref. 16] becomes

$$P_N(\overline{X}, Y, K) = \sum_{b=0}^{\infty} \frac{1}{b!} \frac{\Gamma(K+b)}{\Gamma(K)} \left(\frac{K}{K+\overline{X}_N}\right)^K \left(\frac{\overline{X}_N}{K+\overline{X}_N}\right)^b \sum_{m=0}^{N-1+b} e^{-Y} \frac{Y^m}{m!} \tag{6.1}$$

and Eq. (6.1) can be manipulated as was Eq. (2.3) to obtain

$$P_N(\overline{X}, Y, K) = \sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} + \sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} \left[1 - \sum_{b=0}^{m-N} \frac{(K+b-1)!}{b!\,(K-1)!} \left(\frac{1}{1+\frac{\overline{X}_N}{K}}\right)^K \left(\frac{\frac{\overline{X}_N}{K}}{1+\frac{\overline{X}_N}{K}}\right)^b \right] \tag{6.2}$$

A program to evaluate Eq. (6.2) to within an accuracy ϵ is given in Fig. 8. This program can be modified so as to eliminate underflow-overflow problems as before by combining elements of Figs. 4 and 8 so that the variable K is introduced and the quantities

$$\left(\frac{1}{1+\frac{\overline{X}_N}{K}}\right)^K \text{ and } \left(\frac{\overline{X}_N}{1+\overline{X}_N/K}\right) \frac{K1}{B}$$

replace $e^{-\overline{X}_N}$ and $e^{-\overline{X}_N/K}$, respectively, in Fig. 4.

This formulation of P_D is not as efficient for calculation purposes as is the special expression derived for Cases 0-4.

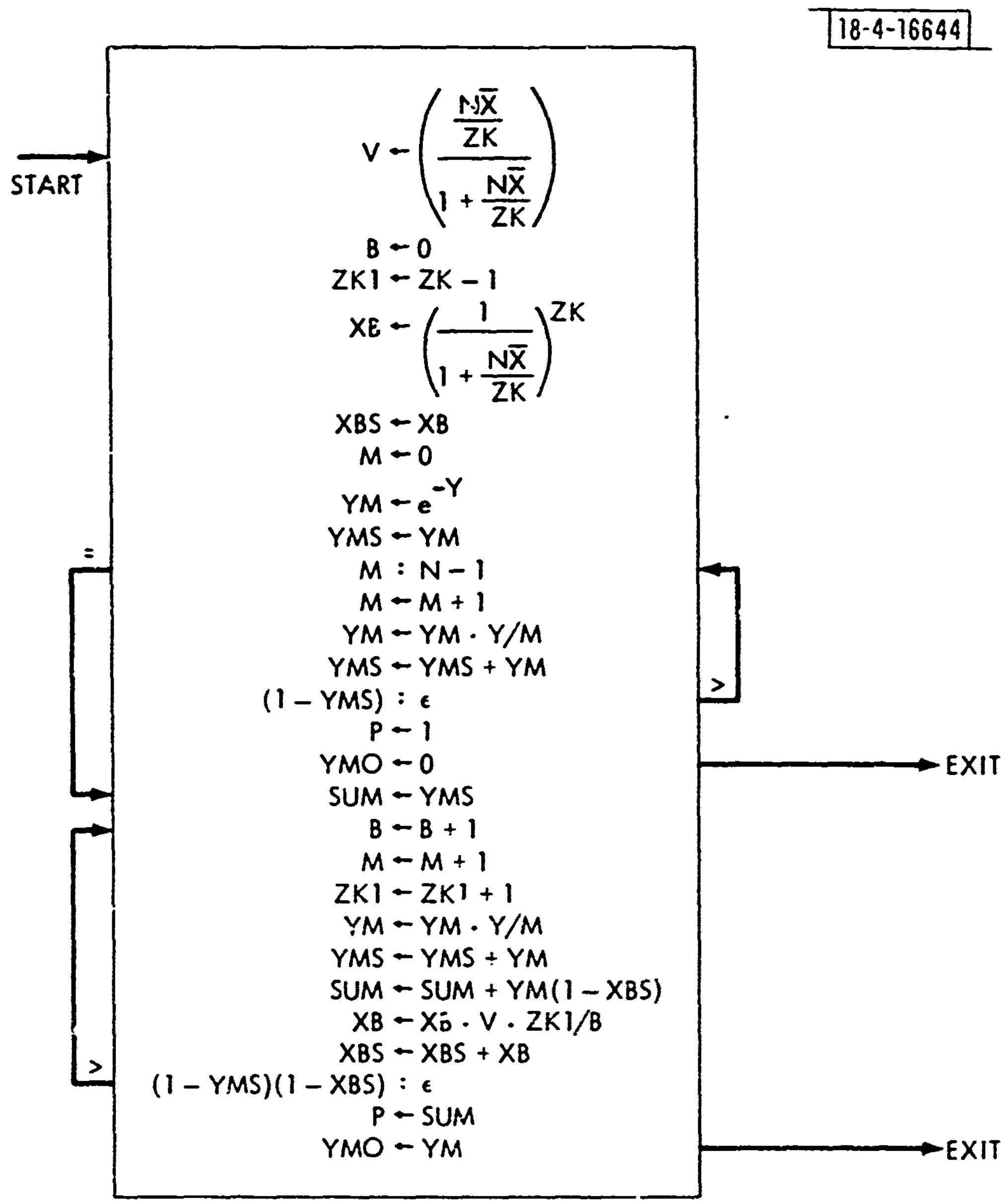

Fig. 8. Program for subroutine PGEN(N, X̄, Y, ZK, P, YMO).

SECTION 7

CONCLUSIONS

Highly efficient and accurate algorithms have been presented for calculating the probability of detection for Marcum (Case 0) and Swerling (Cases 1 through 4) models. Tests and reformulations are included to avoid the underflow-overflow problem usually encountered for large parameter values.

A simple approximation has been provided for the case in which the radar cross section is constant with a scan but fluctuates log-normally from scan to scan. The accuracy is limited but will suffice for most applications. Guidelines as to which parameter combinations might pose an accuracy problem were given.

An efficient algorithm for determining the threshold level Y for a given P_{FA} has also been included.

Finally, with $P_{FA} = 10^{-6}$, Figs. 9, 10, and 11 compared $P_D(s)$ for several of the cases for N = 1, 10, and 100, respectively. Case 4 curves, omitted on Figs. 10 and 11, lie between Case 2 and Case 0. For Case 5, $\rho = 1.5$ is used throughout.

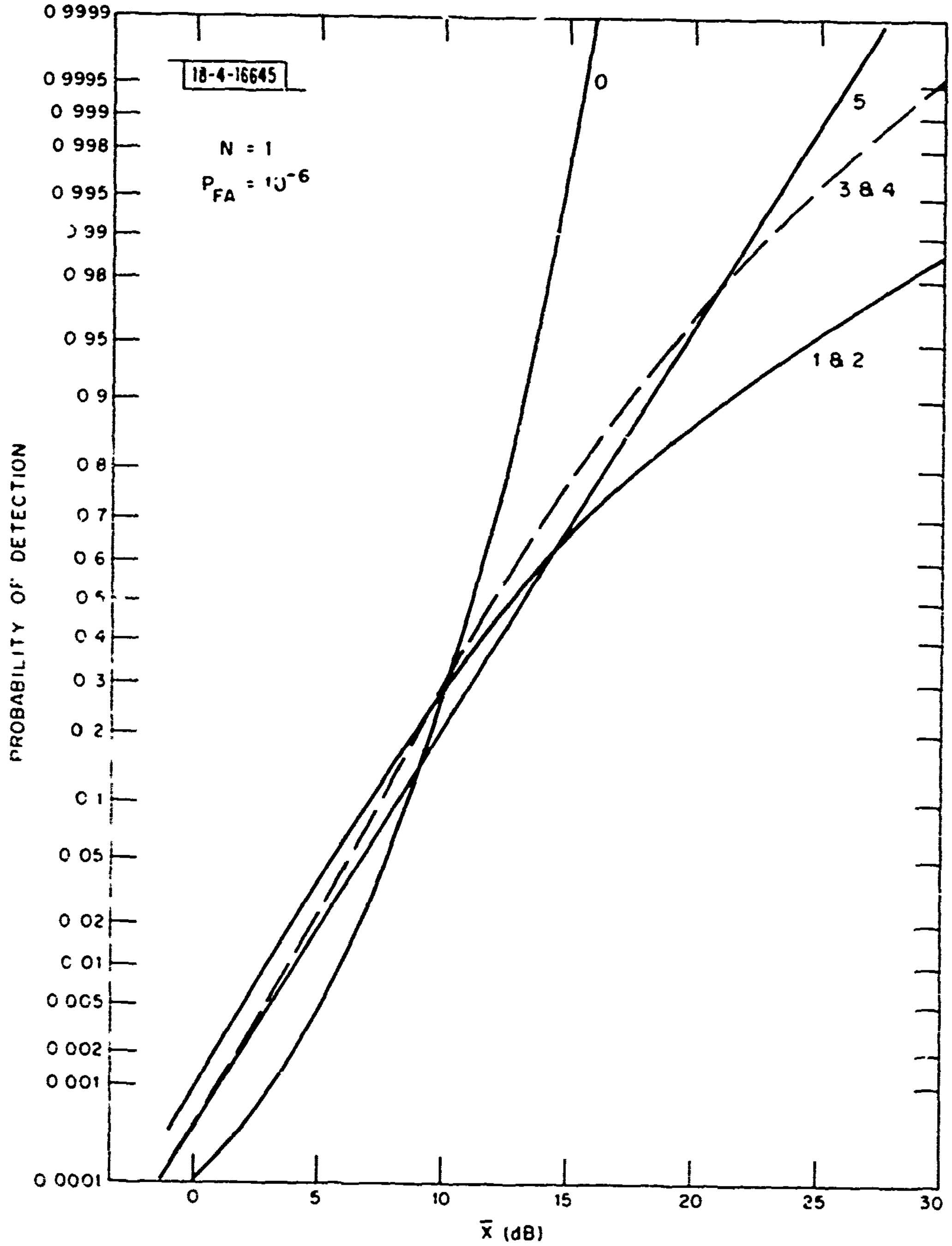

Fig. 9. Probability of detection vs $\overline{X}$ for $N = 1$, $P_{FA} = 10^{-6}$ (Cases 0 – 5).

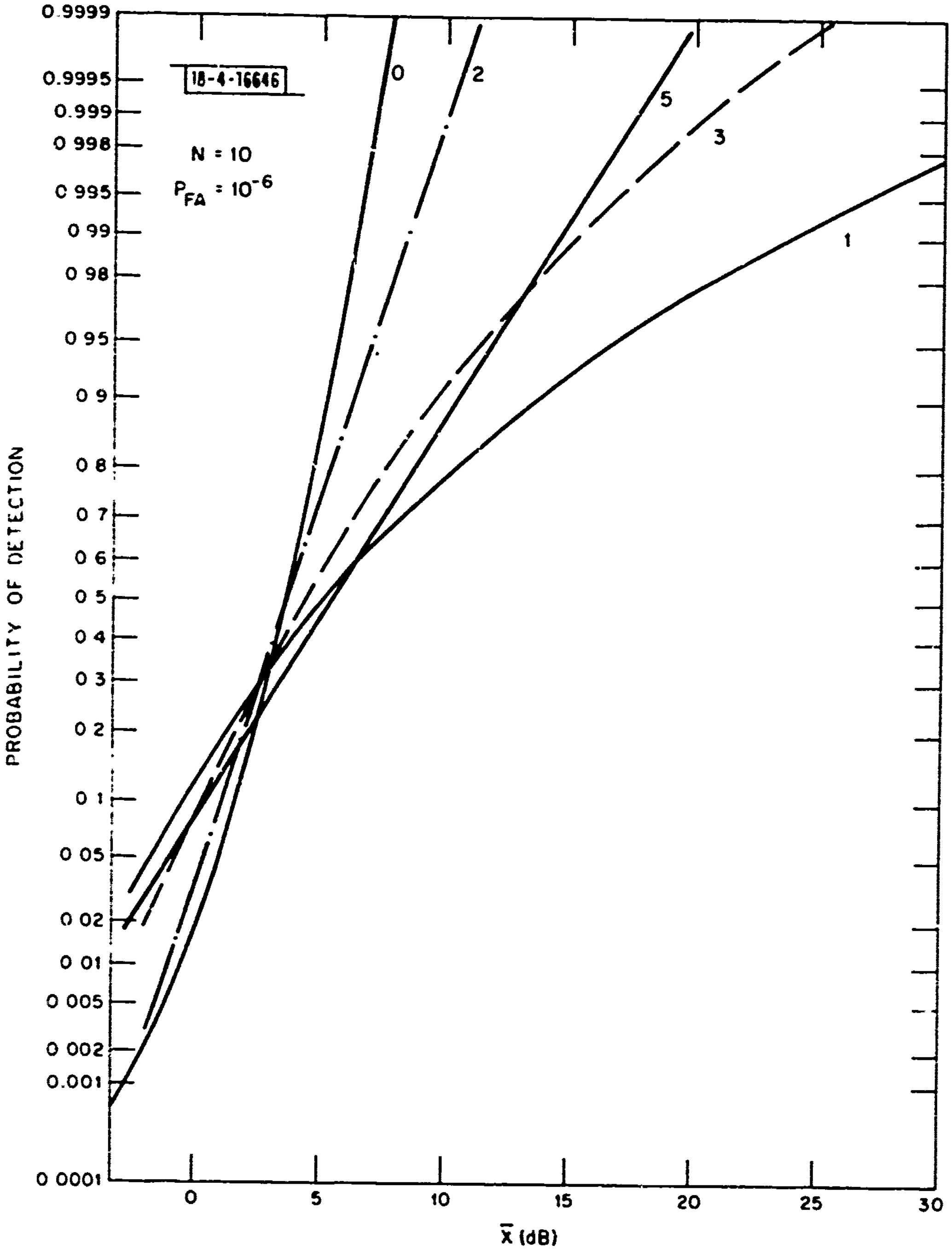

Fig. 10. Probability of detection vs $\bar{X}$ for N = 10, $P_{FA} = 10^{-6}$ (Cases 0 - 3 and 5).

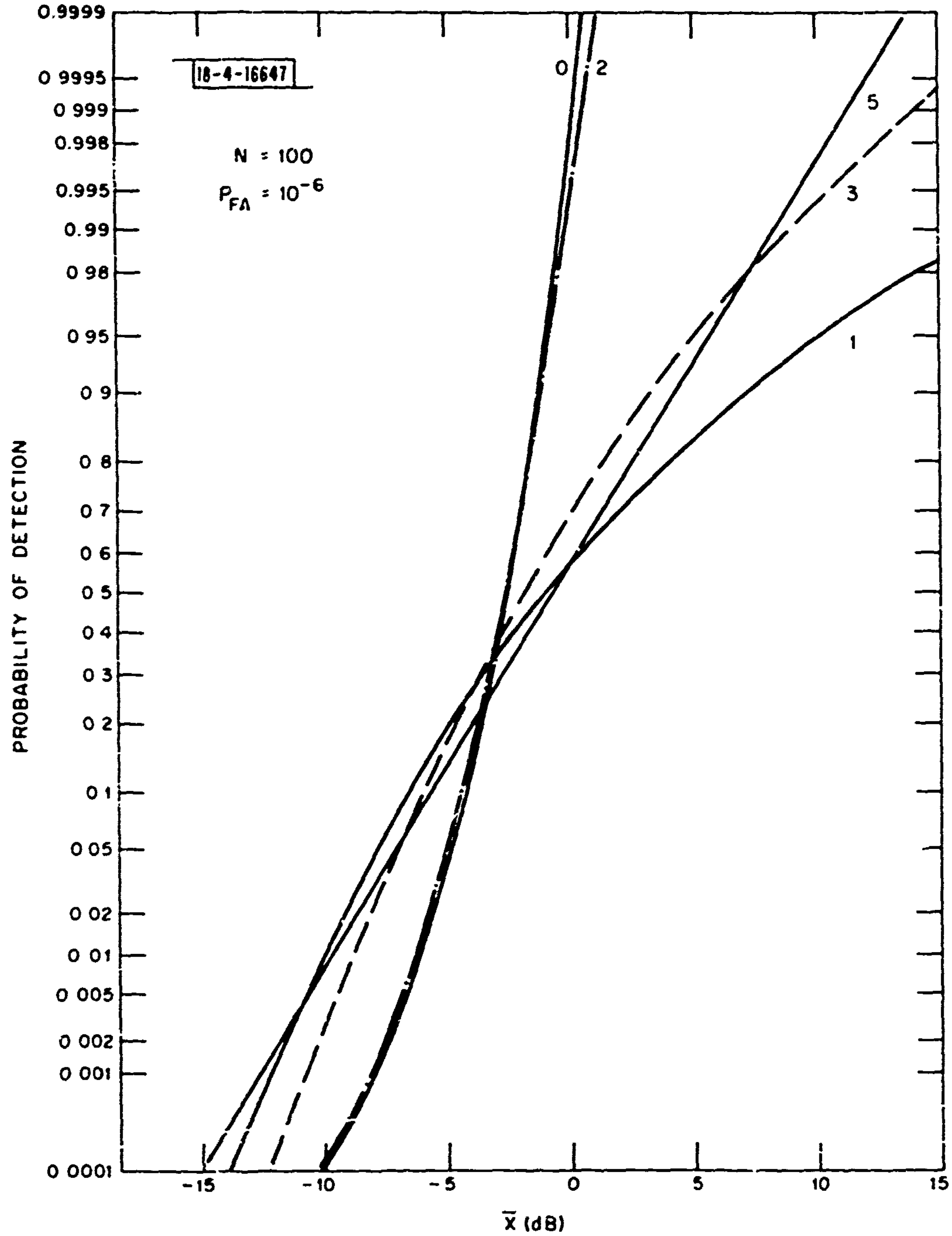

Fig. 11. Probability of detection vs $\overline{X}$ for N = 100, $P_{FA} = 10^{-6}$ (Cases 0 - 3 and 5).

APPENDIX A

FORTRAN PROGRAMS

In this Appendix, FORTRAN versions of the programs to evaluate the probabilities for cases 0-4 for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$ are given. In addition, a program to determine the threshold for a given P_{FA} is given.

The subroutine program for ${}_0P_N(\overline{X}, Y)$ is titled, with $\epsilon = 10^{-6}$, PNXYS(N, X, Y, P, YMO) and, with $\epsilon = 10^{-12}$, PNXYT(N, X, Y, P, YMO) where

N = number of incoherently integrated pulses

X = average single-pulse signal-to-noise ratio

Y = threshold level

P $= {}_0P_N(X, Y)$

$$\text{YMO} = e^{-Y} \frac{Y^{N-1}}{(N-1)!}$$

N, X, Y, are inputs; P and YMO are outputs. The output YMO is included since other programs calling on ${}_0P_N(X, Y)$ have use for it.

Subroutine programs for Swerling (Cases 1-4, are given in a single program with four entry points. The titles for the four cases, with 10^{-6}, are SSWC1(N, XBAR, Y, P), SSWC2(N, XBAR, Y, P), SSWC3(N, XBAR, Y, P) and SSWC4(N, XBAR, Y, P), respectively, and those for $\epsilon = 10^{-12}$ are entitled TSWC1(N, XBAR, Y, P), TSWC2(N, XBAR, Y, P), TSWC3(N, XBAR, Y, P) and TSW4(N, XBAR, Y, P). In all cases we have

N = number of incoherently integrated pulses

XBAR = average single-pulse signal-to-noise ratio

Y = threshold level

P = probability of detection.

A subroutine entitled PREAM (X, CE, X1, X2, X3, X4, XK, IND) is included as it is called by ${}_0P_N(X, Y)$ and ${}_0P_N(X, Y)$. It evaluates the functions $K_\epsilon(X)$ and $M_\epsilon(Y)$ and

X = the value of X or Y

CE = the corresponding C_ϵ of (B:11)

X1 = 18 or 34 for $K_G(X)$, and 40 or 80 for $M_\epsilon(Y)$

X2 = 150 or 175 for $K_\epsilon(X)$

X3 = 0.75 or 0.8 for $K_\epsilon(X)$, and 1.3 or 1.28 for $M_\epsilon(Y)$

X4 = 20 or 55 for $K_\epsilon(X)$, and 24 or 52 for $M_\epsilon(Y)$

IND is a flag, if IND = 2, the routine calculator $M_\epsilon(Y)$, otherwise $K_\epsilon(X)$

XK = output, either $K_\epsilon(X)$ or $M_\epsilon(Y)$

Finally, the subroutine program THRESH (N, TOL, PFA, Y1) is used to determine the threshold Y1 in accordance with Eqs. (5.4) and (5.5) where

N = number of incoherently integrated pulses

TOL = tolerance, i.e., acceptable difference between Y_{K+1} and Y_k for termination of the iterative process

P_{FA} = input false alarm probability

Y1 = output threshold value.

A table of values (Table A-I) is included for the purpose of checking out the programs.

Table A-1

N	Y	Case	$\bar{X}$ = 3.162278	$\bar{X}$ = 10	$\bar{X}$ = 31.62278	$\bar{X}$ = 100	$\bar{X}$=316.2278	$\bar{X}$ = 1000
1	13.81551055	0	.0045853516	.24804931	.9972254	1.	1.	1.
		1	.036181097	.2848026	.67475594	.87215578	.95738396	.9862931
		2	.036181097	.2848036	.65475594	.87215578	.95738396	.9862931
		3	.020265921	.29188211	.77944624	.9652566	.99594141	.99957316
		4	.020265921	.29188211	.77944624	.9652566	.99594141	.99957316
3	19.12916818	0	.088813157	.97272573	1.	1.	1.	1.
		1	.19717145	.57609062	.83647026	.94469186	.98212606	.99430855
		2	.16308152	.7468903	.97820932	.97320932	.99901695	.99999885
		3	.17843655	.68699649	.94510657	.99332715	.99928888	.99992743
		4	.1368032	.83404914	99775036	.99999405	.99999999	1.
10	32.71034051	0	.85331678	1.	1.	1.	1.	1.
		1	.48554348	.79111512	.92802416	.97659606	.99253297	.99763206
		2	.73398702	.99896678	.99999989	1.	1.	1.
		3	.56937472	.91394522	.98900437	.99880847	99987775	.99998767
		4	.78178937	.99996291	1.	1.	1.	1.
30	63.54818012	0	.99999943	1.	1.	1.	1.	1.
		1	.69852634	.98170706	.96429063	.98855538	.99636547	.99884911
		2	.9994473	1.	1.	1.	1.	1.
		3	.83147881	.97590200	.99728587	.99971902	.99997146	.99999713
		4	.99993173	1.	1.	1.	1.	1.
100	154.9190459	0	1.	1.	1.	1.	1.	1.
		1	.83880149	.94571525	.98248263	.99442475	.99823335	.99944098
		2	1.	1.	1.	1.	1.	1.
		3	.94852786	.99385994	.99934996	.9999338	.99999334	.99999933
		4	1.	1.	1.	1.	1.	1.

```
USERID MASR     CLASS A NAME PNXYS       FORTRAN         02/14/75  12028023
OREAD  PNXYS    FORTRAN  A1 MASR    8/20/74  14010
      SUBROUTINE PNXYS(N,X,Y,P,YMO)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA E,EPSILN/175.D0,1.D-6/
      X=DFLOAT(N)*X
      CALL PREAMB(X,16.12D0,18.D0,150.D0,.75D0,20.D0,EKX,1)
      CALL PREAMB(Y,16.12D0,40.D0,40.D0,1.3D0,24.D0,EKY,2)
      IF(DFLOAT(N).LE.EKY-EKX)GO TO 300
      P=1.D0
      YMO=0.D0
      RETURN
300   IB=N-1
      K=0
      IF(X.GT.E)GO TO 10
      XK=DEXP(-X)
    5 XKS=XK
      M=0
      IF(Y.GT.E)GO TO 20
      YM=DEXP(-Y)
   70 YMS=YM
   40 IF(M.EQ.IB)GO TO 30
      M=M+1
      YM=YM*Y/FLOAT(M)
      YMS=YMS+YM
      IF(EPSILN.LE.1.D0-YMS)GO TO 40
      P=YMS
      YMO=YM
      X=X/DFLOAT(N)
      RETURN
   30 SUM=YMS
      YMO=YM
   60 K=K+1
      M=M+1
      YM=YM*Y/FLOAT(M)
      YMS=YMS+YM
91    SUM=SUM+YM*(1.D0-XKS)
      XK=XK*X/FLOAT(K)
      XKS=XKS+XK
      ANS=(1.D0-YMS)*(1.D0-XKS)
      IF(ANS.GT.EPSILN)GO TO 60
   50 P=SUM
      X=X/DFLOAT(N)
      RETURN
20    YMLY=0.D0
      YLOG=DLOG(Y)
  600 M=M+1
      XMF=DLOG(DFLOAT(M))
      YMLY=YMLY+YLOG-XMF
      IF(Y-YMLY.GT.E)GO TO 600
      YM=DEXP(-Y+YMLY)
      IF(M.LE.IB)GO TO 70
      YMO=0.
      YMS=YM
      SUM=0.D0
```

```
      KD=M-N
      IF(DFLOAT(KD).LT.X)GO TO 80
      P=0.D0
      X=X/DFLOAT(N)
      RETURN
   80 K=K+1
      XK=XK*X/FLOAT(K)
      XKS=XKS+XK
      IF(K-KD)80,90,90
   90 K=K+1
      GO TO 91
10    XKLX=0.D0
      XLOG=DLOG(X)
  210 K=K+1
      XKF=DLOG(DFLOAT(K))
      XKLX=XKLX+XLOG-XKF
      IF(X-XKLX.GT.E)GO TO 210
      XK=DEXP(-X+XKLX)
      IB=IB+K
      GO TO 5
      END
```

```
USERID MASR     CLASS A NAME PNXYT        FORTRAN           02/14/75  12028039
OREAD  PNXYT    FORTRAN  A1 MASR    8/20/74  14011
      SUBROUTINE PNXYT(N,X,Y,P,YMO)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA E,EPSILN/175.D0,1.D-12/
      X=DFLOAT(N)*X
      CALL PREAMB(X,30.D0,34.D0,175.D0,.8D0,55.D0,EKX,1)
      CALL PREAMB(Y,30.D0,80.D0,80.D0,1.28D0,52.D0,EKY,2)
      IF(DFLOAT(N).LE.EKY-EKX)GO TO 300
      P=1.D0
      YMO=0.D0
      RETURN
300   IB=N-1
      K=0
      IF(X.GT.E)GO TO 10
      XK=DEXP(-X)
    5 XKS=XK
      M=0
      IF(Y.GT.E)GO TO 20
      YM=DEXP(-Y)
   70 YMS=YM
   40 IF(M.EQ.IB)GO TO 30
      M=M+1
      YM=YM*Y/FLOAT(M)
      YMS=YMS+YM
      IF(EPSILN.LE.1.D0-YMS)GO TO 40
      P=YMS
      YMO=YM
      X=X/DFLOAT(N)
      RETURN
   30 SUM=YMS
      YMO=YM
   60 K=K+1
      M=M+1
      YM=YM*Y/FLOAT(M)
      YMS=YMS+YM
71    SUM=SUM+YM*(1.D0-XKS)
      XK=XK*X/FLOAT(K)
      XKS=XKS+XK
      ANS=(1.D0-YMS)*(1.D0-XKS)
      IF(ANS.GT.EPSILN)GO TO 60
   50 P=SUM
      X=X/DFLOAT(N)
      RETURN
20    YMLY=0.D0
      YLOG=DLOG(Y)
  600 M=M+1
      XMF=DLOG(DFLOAT(M))
      YMLY=YMLY+YLOG-XMF
      IF(Y-YMLY.GT.E)GO TO 600
      YM=DEXP(-Y+YMLY)
      IF(M.LE.IB)GO TO 70
      YMO=0.
      YMS=YM
      SUM=0.D0
```

```
      KD=M-N
      IF(DFLOAT(KD).LT.X)GO TO 80
      P=0.D0
      X=X/DFLOAT(N)
      RETURN
   80 K=K+1
      XK=XK*X/FLOAT(K)
      XKS=XKS+XK
      IF(K-KD)80,90,90
   90 K=K+1
      GO TO 91
10    XKLX=0.D0
      XLOG=DLOG(X)
  210 K=K+1
      XKF=DLOG(DFLOAT(K))
      XKLX=XKLX+XLOG-XKF
      IF(X-XKLX.GT.E)GO TO 210
      XK=DEXP(-X+XKLX)
      IB=IB+K
      GO TO 5
      END
```

```
USERID MASR      CLASS A NAME SSCASE       FORTRAN             02/14/75  12029024
OREAD  SSCASE    FORTRAN  A1 MASR     8/22/74  12049
      SUBROUTINE SSWC1(N,XBAR,Y,P)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA E,EPSILN/175.D0,1.D-6/
      XBAR=DFLOAT(N)*XBAR
      CONST=Y/(1.D0+XBAR)
      IF(N.GT.1)GO TO 5
      IF(CONST.GT.E)GO TO 6
      P=DEXP(-CONST)
      GO TO 100
    6 P=0.D0
      GO TO 100
    5 SUM=0.D0
      EX=CONST-DFLOAT(N-1)*DLOG(1.D0+1.D0/XBAR)
      IF(EX.GE.30.D0)GO TO 10
      CALL PNXYS(N-1,0.D0,Y/(1.+1./XBAR),PP,YSUM)
      SUM=(1.D0-PP)*DEXP(-EX)
   10 CALL PNXYS(N-1,0.D0,Y,P1,YSUM)
      P=SUM+P1
  100 XBAR=XBAR/DFLOAT(N)
      RETURN
      ENTRY SSWC2(N,XBAR,Y,P)
      CONST=Y/(1.D0+XBAR)
      CALL PNXYS(N,0.D0,CONST,P,YSUM)
      RETURN
      ENTRY SSWC3(N,XBAR,Y,P)
      XBAR=DFLOAT(N)*XBAR
      C3=1.D0+XBAR/2.D0
      CONST=Y/C3
      C2=1.D0+2.D0/XBAR
      IF(N.EQ.1)GO TO 20
      SUM=0.D0
      EX=CONST-DFLOAT(N-2)*DLOG(C2)
      C=DEXP(-EX)*(1.D0-2.D0*DFLOAT(N-2)/XBAR+Y/C3)
      IF(C.LE.EPSILN)GO TO 200
      CALL PNXYS(N-1,0.D0,Y/C2,P1,YSUM)
      SUM=C*(1.D0-P1)
  200 CALL PNXYS(N-1,0.D0,Y,P2,YSUM)
      SUM=SUM+P2
      P=SUM+YSUM*1.D0/C3*Y
      GO TO 250
   20 IF(CONST.LT.E)GO TO 21
      P=0.D0
      GO TO 250
   21 P=DEXP(-CONST)*(1.D0+XBAR/(2.D0*C3)*Y/C3)
  250 XBAR=XBAR/DFLOAT(N)
      RETURN
      ENTRY SSWC4(N,XBAR,Y,P)
      C3=1.D0+XBAR/2.D0
      XN=DFLOAT(N)*DLOG(C3)
      CONST=Y/C3
      CALL PNXYS(N,0.D0,CONST,SUM,YSUM)
      IF(1.D0-SUM.GT.EPSILN)GO TO 30
      P=1.D0
```

```
      GO TO 300
   30 CALL PREAMB(CONST,16.12D0,18.D0,150.D0,.75D0,20.D0,EKY,1)
      IF(DFLOAT(2*N-1).GE.EKY)GO TO 31
      P=0.D0
      GO TO 300
   31 M=N-1
      IF(XN.GT.E)GO TO 40
      ZK=DEXP(-XN)
   51 ZKS=ZK
      Z=XBAR/2.D0
   42 IF(M.EQ.2*N-1)GO TO 41
      M=M+1
      YSUM=YSUM*CONST/DFLOAT(M)
      SUM=SUM+YSUM*(1.D0-ZKS)
      ZK=ZK*Z*(2.D0*DFLOAT(N)-DFLOAT(M))/(DFLOAT(M)-DFLOAT(N)+1.D0)
      ZKS=ZKS+ZK
      GO TO 42
   41 P=SUM
300   RETURN
   40 M=M+1
      XN=XN-DLOG(XBAR/2.D0)-DLOG(2.D0*DFLOAT(N)-DFLOAT(M))/(DFLOAT(M)-DF
     1LOAT(N)+1)
      IF(XN.GT.E)GO TO 50
      ZK=DEXP(-XN)
      GO TO 51
   50 YSUM=YSUM*CONST/DFLOAT(M)
      SUM=SUM+YSUM
      GO TO 40
      END
```

```
USERID MASR     CLASS A NAME TSCASE      FORTRAN          02/14/75  12029042
OREAD  TSCASE   FORTRAN  A1 MASR    8/29/74  11014
      SUBROUTINE TSWC1(N,XBAR,Y,P)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA E,EPSILN/175.D0,1.D-12/
      XBAR=DFLOAT(N)*XBAR
      CONST=Y/(1.D0+XBAR)
      IF(N.GT.1)GO TO 5
      IF(CONST.GT.E)GO TO 6
      P=DEXP(-CONST)
      GO TO 100
    6 P=0.D0
      GO TO 100
    5 SUM=0.D0
      EX=CONST-DFLOAT(N-1)*DLOG(1.D0+1.D0/XBAR)
      IF(EX.GE.30.D0)GO TO 10
      CALL PNXYT(N-1,0.D0,Y/(1.+1./XBAR),PP,YSUM)
      SUM=(1.D0-PP)*DEXP(-EX)
   10 CALL PNXYT(N-1,0.D0,Y,P1,YSUM)
      P=SUM+P1
100   XBAR=XBAR/DFLOAT(N)
      RETURN
      ENTRY TSWC2(N,XBAR,Y,P)
      CONST=Y/(1.D0+XBAR)
      CALL PNXYT(N,0.D0,CONST,P,YSUM)
      RETURN
      ENTRY TSWC3(N,XBAR,Y,P)
      XBAR=DFLOAT(N)*XBAR
      C3=1.D0+XBAR/2.D0
      CONST=Y/C3
      C2=1.D0+2.D0/XBAR
      IF(N.EQ.1)GO TO 20
      SUM=C.D0
      EX=CONST-DFLOAT(N-2)*DLOG(C2)
      C=DEXP(-EX)*(1.D0-2.D0*DFLOAT(N-2)/XBAR+Y/C3)
      IF(C.LE.EPSILN)GO TO 200
      CALL PNXYT(N-1,0.D0,Y/C2,P1,YSUM)
      SUM=C*(1.D0-P1)
  200 CALL PNXYT(N-1,0.D0,Y,P2,YSUM)
      SUM=SUM+P2
      P=SUM+YSUM*1.D0/C3*Y
      GO TO 250
   20 IF(CONST.LT.E)GO TO 21
      P=0.D0
      GO TO 250
   21 P=DEXP(-CONST)*(1.D0+XBAR/(2.D0*C3)*Y/C3)
250   XBAR=XBAR/DFLOAT(N)
      RETURN
      ENTRY TSWC4(N,XBAR,Y,P)
      C3=1.D0+XBAR/2.D0
      XN=DFLOAT(N)*DLOG(C3)
      CONST=Y/C3
      CALL PNXYT(N,0.D0,CONST,SUM,YSUM)
      IF(1.D0-SUM.GT.EPSILN)GO TO 30
      P=1.D0
```

```
      GO TO 300
   30 CALL PREAMB(CONST,30.D0,34.D0,175.D0,.8D0.55.D0,EKY,1)
      IF(DFLOAT(2*N-1).GE.EKY)GO TO 31
      P=0.D0
      GO TO 300
   31 M=N-1
      IF(XN.GT.E)GO TO 40
      ZK=DEXP(-XN)
   51 ZKS=ZK
      Z=XBAR/2.D0
   42 IF(M.EQ.2*N-1)GO TO 41
      M=M+1
      YSUM=YSUM*CONST/DFLOAT(M)
      SUM=SUM+YSUM*(1.D0-ZKS)
      ZK=ZK*Z*(2.D0*DFLOAT(N)-DFLOAT(M))/(DFLOAT(M)-DFLOAT(N)+1.D0)
      IF(MOD(M,9).NE.0)GO TO 47
      WRITE(6,48)ZK
   48 FORMAT(1X,'ZK=',D15.8)
47    ZKS=ZKS+ZK
      GO TO 42
   41 P=SUM
300   RETURN
   40 M=M+1
      XN=XN-DLOG(XBAR/2.D0)-DLOG(2.D0*DFLOAT(N)-DFLOAT(M))/(DFLOAT(M)-DF
     1LOAT(N)+1)
      IF(XN.GT.E)GO TO 50
      ZK=DEXP(-XN)
      GO TO 51
   50 YSUM=YSUM*CONST/DFLOAT(M)
      SUM=SUM+YSUM
      GO TO 40
      END
```

```
USERID MASR     CLASS A NAME PREAMB       FORTRAN          02/14/75  12028008
OREAD  PREAMB   FORTRAN  A1 MASR    8/20/74  15016
      SUBROUTINE PREAMB(X,CE,X1,X2,X3,X4,XK,IND)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA TWOPI/6.2831853D0/
      IF(IND.EQ.2)GO TO 20
      IF(X.GT.X1)GO TO 1
      XK=0.D0
      RETURN
1     IF(X.GT.X1.AND.X.LT.X2)GO TO 21
      XK=X3*X-X4
      RETURN
   21 CONST=-1.D0
    2 XK=X+CONST*DSQRT(CE*X)
    5 FX=X-XK+XK*DLOG(XK/X)+.5*DLOG(TWOPI*XK)-CE
      FXP=DLOG(XK/X)+.5D0/XK
      XK1=XK-FX/FXP
      IF(DABS(XK1-XK).LT..1D0)RETURN
      XK=XK1
      GO TO 5
   20 CONST=1.D0
      IF(X.LT.X1)GO TO 2
      XK=X3*X+X4
      RETURN
      END
```

```
USERID MASR     CLASS A NAME THRES        FORTRAN           02/14/75  12028014
OREAD  THRES    FORTRAN  A1 MASR    8/22/74  15025
      SUBROUTINE THRESH(N,TOL,PFA,Y1)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA E/175.D0/
      PLOG=DLOG10(PFA)
      XPFA=1.D0/PFA
      IF(N.GE.40.D0)GO TO 30
      Y=DFLOAT(N)**1.15-2.D0*PLOG
      GO TO 31
   30 Y=DFLOAT(N)-8.D0*PLOG
   31 IF(Y.GT.E)GO TO 10
  212 M=0
      YLX=0.D0
      YLOG=DLOG(Y)
      TSUM=DEXP(-Y)
   21 IF(M.EQ.N-1)GO TO 20
  110 M=M+1
      SUMK=DLOG(DFLOAT(M))
      YLX=YLX+YLOG-SUMK
      TSUM=TSUM+DEXP(-Y+YLX)
      GO TO 21
   20 Y1=Y+DLOG(XPFA*TSUM)*TSUM/DEXP(-Y+YLX)
      IF(DABS(Y1-Y).LE.TOL)RETURN
      Y=Y1
      GO TO 31
   10 M=0
      YLX=0.D0
      YLOG=DLOG(Y)
   11 M=M+1
      SUMK=DLOG(DFLOAT(M))
      YLX=YLX+YLOG-SUMK
      IF(Y-YLX.GT.E)GO TO 11
      TSUM=DEXP(-Y+YLX)
      IF(M-N+1)110,20,20
      END
      IMPLICIT REAL*8(A-G,O-Z)
 1000 READ(5,*)N,TOL,PFA,IEND
      CALL THRESH(N,TOL,PFA,Y)
      WRITE(6,100)Y
  100 FORMAT(1X,'Y=',D15.8)
      IF(IEND.EQ.0)RETURN
      GO TO 1000
      END
```

APPENDIX B

DETERMINATION OF THE FUNCTIONS $K_\epsilon(X)$ AND $M_\epsilon(Y)$

The function, $K_\epsilon(X)$, is defined to have the property that for all integers K such that

$$K < K_\epsilon(X) \tag{B.1}$$

then

$$\sum_{k=0}^{K} e^{-X} \frac{X^k}{k!} < \epsilon/2 \tag{B.2}$$

$K_\epsilon(X) = 0$ denotes that the sum is empty.

$M_\epsilon(Y)$ is defined to have the property that for all integers N such that

$$N \geq M_\epsilon(Y) \tag{B.3}$$

then

$$\sum_{m=0}^{N-1} e^{-Y} \frac{Y^m}{m!} > 1 - \epsilon/2 \tag{B. 4}$$

or equivalently

$$\sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} < \epsilon/2 \quad . \tag{B. 5}$$

Using the square bracket notation to mean the largest integer less than or equal to the quantity in the bracket, then ideally, we should determine $K_\epsilon(X)$ such that $[K_\epsilon(X)]$ is the minimum integer for which Eq. (B. 2) is true, and $[M_\epsilon(Y)]$ is the maximum integer for which Eq. (B. 4) is true; it is of little practical significance to have the exact values. Values close to their minimum and maximum, respectively, are much easier to derive.

In order to derive a useful $K_\epsilon(X)$, we begin by noting that

$$\begin{aligned}
\sum_{k=0}^{K} e^{-X} \frac{X^k}{k!} &= e^{-X} \frac{X^K}{K!} \left[1 + \frac{K}{X} + \frac{K(K-1)}{X^2} + \ldots + \frac{K!}{X^k}\right] \\
&\le e^{-X} \frac{X^K}{K!} \left[1 + \frac{K}{X} + \left(\frac{K}{X}\right)^2 + \ldots + \left(\frac{K}{X}\right)^K\right] \\
&= e^{-X} \frac{X^K}{K!} \left[\frac{1 - \left(\frac{K}{X}\right)^{K+1}}{1 - \frac{K}{X}}\right] \\
&\le e^{-X} \frac{X^K}{\sqrt{2\pi K}\left(\frac{K}{e}\right)^k} \frac{1}{1 - \frac{K}{X}}
\end{aligned} \tag{B. 6}$$

where Sterling's approximation has been used for $K!$. If we restrict, for example, $K/X < 4/5$, then Eq. (B. 6) indicates that Eq. (B. 2) is satisfied if

$$5e^{-X} \frac{X^K}{\sqrt{2\pi K}\,\frac{K}{e}^K} \leq \epsilon/2 \tag{B. 7}$$

or equivalently

$$-e^{-\left(X - K + K\ell n \frac{K}{X} + \frac{1}{2} \ell n\, 2\pi K\right)} \leq \epsilon/10 \quad . \tag{B. 8}$$

For $\epsilon = 10^{-6}$ we have

$$\frac{\epsilon}{10} > e^{-16.12} \tag{B. 9}$$

and for $\epsilon = 10^{-12}$

$$\frac{\epsilon}{10} > e^{-30.0} \tag{B. 10}$$

so that if we solve

$$f_X(K,\epsilon) \equiv X - K + K\,\ell n \frac{K}{X} + \frac{1}{2}\,\ell n\, 2\pi K - C_\epsilon = 0 \tag{B. 11}$$

for K, where

$$C_\epsilon = \begin{cases} 16.12 & \text{for } \epsilon = 10^{-6} \\ 30.0 & \text{for } \epsilon = 10^{-12} \end{cases} ,$$

the solution can be used for $K_\epsilon(X)$ provided $\frac{K}{X}$ is less than 4/5. The Newton-Raphson iteration technique to solve Eq. (B.11) is

$$K_{n+1} = K_n - \frac{f_X(K_n, \epsilon)}{f'_X(K_n, \epsilon)} \tag{B.12}$$

where

$$f'_X(K_n, \epsilon) = \ln\left(\frac{K_n}{X}\right) + \frac{1}{2K_n} \quad . \tag{B.13}$$

The initial value for K is obtained for Eq. (B.11) by approximating $\ln Z$ by $Z - 1$ and ignoring the term $\frac{1}{2} \ln 2\pi K$. Eq. (B.11) then becomes

$$X - K + K\left(\frac{K}{X} - 1\right) - C_\epsilon = 0$$

or

$$K^2 - 2XK + X^2 - C_\epsilon X = 0 \quad .$$

Solving for K

$$K = X \pm \sqrt{X^2 - X^2 + C_\epsilon X} = X \pm \sqrt{C_\epsilon X} \quad . \tag{B. 14}$$

Since we desire the smaller of the two roots of $f_x(C, \epsilon)$, we choose the negative sign for K_0

$$K_0 = X - \sqrt{C_\epsilon X} \quad . \tag{B. 15}$$

In Table B-I, empirically determined values of K, for which Eq. (B. 2) is satisfied, are given for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$.

Table B-I

$\epsilon = 10^{-6}$ X	K
18	1
30	6
60	25
100	54
150	93
200	134
250	176
300	218

$\epsilon = 10^{-12}$ X	K
32	1
60	13
100	37
150	70
175	88
200	107
300	184
350	224
400	265
500	348
600	433
700	519

From this table we have, with $\epsilon = 10^{-6}$, that for $X \leq 300$, then $\frac{K}{X} \leq \frac{218}{300} = 0.727 < 0.8$; and with $\epsilon = 10^{-12}$, that for $X \leq 700$, then $\frac{K}{X} \leq \frac{519}{700} = 0.741 < 0.8$. For these values of X we can use

$$K_\epsilon(X) = \text{root of } f_X(K,\epsilon) \quad . \tag{B. 16}$$

For larger values of X, we need another expression. By curve fitting, we obtain for $K_\epsilon(X)$

$$K_\epsilon(X) = 0.75X - 20 \tag{B. 17}$$

for $X > 150$. Since Eq. (B. 17) is much more simple than Eq. (B. 16), we use it down to $X > 150$. For $\epsilon = 10^{-12}$, we similarly obtain

$$K_\epsilon(X) = 0.8\,X - 55 \tag{B. 18}$$

for $X > 175$. For $X < 18$ with $\epsilon = 10^{-6}$ and for $X < 34$ with $\epsilon = 10^{-12}$, we set $K_\epsilon(X)$ to zero. Summarizing our results we have

$$K_{10^{-6}}(X) = \begin{cases} 0 & \text{for } X < 34 \\ \text{root of } f_X(K,\ 10^{-6}) & \text{for } 18 \leq X \leq 150 \\ 0.75\,X - 20 & \text{for } 150 < X \end{cases} \tag{B. 19}$$

$$K_{10^{-12}}(X) = \begin{cases} 0 & \text{for } X < 34 \\ \text{root of } f_X(K,\ 10^{-12}) & \text{for } 34 \le X \le 175\,. \\ 0.8\,X - 55 & \text{for } 175 < X \end{cases} \quad \text{(B. 20)}$$

$M_\epsilon(Y)$ is dealt with in a similar manner to that of $K_\epsilon(X)$. If $N > M_\epsilon(Y)$, then $Y/N < 1$, we have

$$\sum_{m=N}^{\infty} e^{-Y} \frac{Y^m}{m!} = e^{-Y} \frac{Y^N}{N!} \left[1 + \frac{Y}{N+1} + \frac{Y^2}{(N+1)(N+2)} + \dots\right]$$

$$\le e^{-Y} \frac{Y^N}{N!} \left[1 + \frac{Y}{N} + \left(\frac{Y}{N}\right)^2 + \dots\right]$$

$$= e^{-Y} \frac{Y^N}{N!} \left(\frac{1}{1 - \frac{Y}{N}}\right) \le e^{-Y} \frac{Y^N}{\sqrt{2\pi N}\left(\frac{N}{e}\right)^N} \frac{1}{\left(1 - \frac{Y}{N}\right)}\,.$$

(B. 21)

If

$$e^{-Y} \frac{Y^N}{\sqrt{2\pi N}\,\frac{N}{e}^N} \frac{1}{1 - Y/N} \le \epsilon/2 \quad \text{(B. 22)}$$

then Eqs. (B. 4) and (B. 5) are satisfied. If $Y/N < 4/5$, then

$$5\, e^{-Y} \frac{Y^N}{\sqrt{2\pi N}\left(\frac{N}{e}\right)^N} \le \epsilon/2$$

or equivalently

$$e^{-\left[Y - N + N\, \ell n(N/X) + \frac{1}{2}\, \ell n\, 2\pi K\right]} \le \epsilon/10 \tag{B. 23}$$

which reduces as in Eq. (B. 11) to

$$f_Y(N, \epsilon) = 0 \quad . \tag{B. 24}$$

An empirically determined table of values of $M_\epsilon(Y)$ are given in Table B-II.

Table B-II

$\epsilon = 10^{-6}$

Y	N
1	11
10	30
20	46
40	76
50	89
80	129
100	154
150	218

$\epsilon = 10^{-12}$

Y	N
1	15
10	41
20	60
40	94
80	153
100	180
175	278
250	372
400	554

Although Eq. (B. 24) is the same as Eq. (B. 11) except that we have replaced X by Y and K by N, we are not looking for the same root. $f_Y(N,\epsilon)$ has two roots and this time we need the larger root; whereas for $f_X(K,\epsilon)$ we needed the smaller. This is accounted for by our initial value for N, which uses the plus sign in Eq. (B. 14)

$$N_0 = Y + \sqrt{C_\epsilon Y} \quad . \tag{B. 25}$$

The root can be used for large values of Y, but since we have a more simple expression

$$M_{10^{-6}}(Y) = 1.3\ Y + 24 \qquad \text{for } Y \geq 40$$

and

$$M_{10^{-12}}(Y) = 1.28\ Y + 52 \qquad \text{for } Y \geq 80$$

we use

$$M_{10^{-6}}(Y) = \begin{cases} \text{root of } f_N(Y, 10^{-6}) & \text{for } Y < 40 \\ 1.3\ Y + 24 & \text{for } Y \geq 40 \end{cases}$$

and

$$M_{10^{-12}}(Y) = \begin{cases} \text{root of } f_N(Y, 10^{-12}) & \text{for } Y < 80 \\ 1.28\ Y + 52 & \text{for } Y \geq 80 \end{cases} .$$

APPENDIX C

ASYMPTOTIC VALUE OF ${}_0P_N\left(\frac{Y - (N - 1)}{N},\ Y\right)$

We show that

$$\lim_{Y \to \infty} \ {}_0P_N\left(\frac{Y - (N - 1)}{N}\ ,\ Y\right) = \frac{1}{2} \quad . \tag{C. 1}$$

Gradshten and Ryzhik [Ref. 17] (p. 717, 6. 63. 8) give us

$$\int_0^1 z^n e^{-\gamma z^2} I_{n-1}(2\gamma z)\, dz = \frac{1}{4\gamma}\left[e^{\gamma} - e^{-\gamma} \sum_{k=-(n-1)}^{(n-1)} I_k(2\gamma)\right] . \tag{C. 2}$$

From Eq. (2. 1) we have

$$1 - {}_0P_N(X, Y) = \int_0^Y \left(\frac{v}{NX}\right)^{\frac{N-1}{2}} e^{-(v + NX)}\, I_{N-1}(2\sqrt{NXv})\ dv \tag{C. 3}$$

so that by change of variables $\alpha^2 Z^2 = v$ and $\alpha^2 = NX$, we obtain

$$1 - {}_0P_N(X, Y) = \frac{2\alpha^{N+1}}{\alpha^{N-1}} \int_0^{\frac{\sqrt{Y}}{\alpha}} Z^N e^{-(\alpha^2 Z + \alpha^2)} I_{N-1}(2\alpha^2 Z)\ dZ$$

$$= 2\alpha^2 e^{-\alpha^2} \int_0^{\frac{\sqrt{Y}}{\alpha}} Z^N e^{-\alpha^2 Z} I_{N-1}(2\alpha^2 Z)\ dZ$$

$$= 2\alpha^2 e^{-\alpha^2} \int_0^{1} Z^N e^{-\alpha^2 Z} I_{N-1}(2\alpha^2 Z)\ dZ$$

$$+ 2\alpha^2 e^{-\alpha^2} \int_1^{\frac{\sqrt{Y}}{\alpha}} Z^N e^{-\alpha^2 Z} I_{N-1}(2\alpha^2 Z)\ dZ\ . \qquad \text{(C. 4)}$$

With $\alpha^2 = Y - (N - 1)$ and using Eq. (C. 2) we obtain

$$1 - {}_0P_N\left(\frac{Y - (N-1)}{N},\ Y\right)$$

$$= 2\alpha^2 e^{-\alpha^2} \frac{1}{4\alpha^2} \left[e^{\alpha^2} - e^{-\alpha^2} \sum_{k=-(N-1)}^{N-1} I_k(2\alpha^2) \right]$$

$$+ 2\alpha^2 e^{-\alpha^2} \int_1^{\sqrt{\frac{Y}{Y-(N-1)}}} Z^N e^{-\alpha^2 Z^2} I_{N-1}(2\alpha^2 Z)\ dZ$$

$$= \frac{1}{2}\left[1 - e^{-2[Y-(N-1)]} \sum_{k=-(N-1)}^{N-1} I_k(2[Y-(N-1)])\right]$$

$$+ 2[Y-(N-1)]\, e^{-[Y-(N-1)]} \int_1^{\sqrt{\frac{Y}{Y-(N-1)}}} Z^N e^{-[Y-(N-1)]} I_N(2[Y-(N-1)]Z)\, dZ \; . \tag{C.5}$$

Taking the limit as $Y \to \infty$

$$\sqrt{\frac{Y}{Y-(N-1)}} \to 1 \tag{C.6}$$

and

$$e^{-2[Y-(N-1)]} I_k(2[Y-(N-1)]) \to 0 \text{ for } |K| \le N-1 \tag{C.7}$$

so that Eq. (C.1) is obtained.

APPENDIX D

CASE 5 FORTRAN PROGRAM

The approximation is simply

$$ {}_5P_N(\overline{X}, Y, \rho) \approx \frac{1}{2} \operatorname{erfc}\left[\frac{\ln\left[\frac{Y - (N - 1)}{NM}\right]}{\sqrt{2}\sigma}\right] \qquad (D.1) $$

where

$$ M = \overline{X}/\rho \qquad (D.2) $$

and

$$ \sigma = 2\sqrt{\ln\rho} \quad . \qquad (D.3) $$

The subroutine program titled PNLN (N, X, Y, R, P5) is given below where

N = Number of pulses incoherently integrated

X = Average Input Signal-to-Noise Ratio

Y = Threshold Level

R = ρ

P5 = $P_N(X, Y, \rho)$.

The complementary error function subroutine ERFC (X, ERC) called by PNLN(N, X, Y, R, P5) also given below is based on Hastings [18] algorithm for the error function.

```
USERID MASR     CLASS A NAME PNLN        FORTRAN          02/14/75  12029019
OREAD  PNLN     FORTRAN  A1 MASR    8/27/74  14039
      SUBROUTINE PNLN(N,X,Y,RHO,P5)
      XM=X/RHO
      SIG= SQRT(2.  *ALOG(RHO))
      ARG=ALOG((Y- FLOAT(N-1))/( FLOAT(N)*XM))/(1.414213562  *SIG)
      CALL EREC( ARG, ERC)
      P5=.5  *ERC
      RETURN
      END
      SUBROUTINE EREC(X,ERC)
      DATA A1,A2,A3,A4,A5,P,PI/.225836846,-.252128668,1.25969513,
     1-1.287822453,.94064607,.3275911,3.14159265/
      SPI=SQRT(PI)
      IF(X)10,2,2
    2 C1=0.0
      C2=1.0
    3 ETA=1./(1.+P*X)
      POLY=(((((A5*ETA+A4)*ETA+A3)*ETA+A2)*ETA+A1)*ETA)
      ERC=C1+C2*2.*POLY*EXP(-X**2)/SPI
      RETURN
   10 C1=2.
      C2=-1.
      X=-X
      GO TO 3
      END
```

ACKNOWLEDGMENT

The author wishes to express his gratitude to Barbara Forman, whose conscientious effort in writing the programs was of great assistance, and to Darrol DeLong, whose comments and corrections were greatly appreciated.

REFERENCES

[1] J. I. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar," Trans. IRE, PGIT IT-6, pp. 59-144 (1960).

[2] J. I. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar: Mathematical Appendix," Trans. IRE, PGIT IT-6, pp. 145-268 (1960).

[3] P. Swerling, "Probability of Detection for Fluctuating Targets," Trans. IRE, PGIT IT-6, pp. 269-308 (1960).

[4] L. F. Fehlner, "Marcum's and Swerling's Data on Target Detection by a Pulsed Radar," Appl. Physics Lab/Johns Hopkins University, Dept. TG-451, Silver Spring, MD, July 1962.

[5] J. V. DiFranco and W. L. Rubin, Radar Detection (Prentice-Hall, New Jersey, 1968).

[6] D. P. Meyer and H. A. Mayer, Radar Target Detection (Academic Press, New York, 1965).

[7] G. R. Heidbreder and R. L. Mitchell, "Detection Probabilities for Log-Normally Distributed Signals," IEEE Trans. Aerospace Electron. Syst., AES-3, pp. 5-13 (1967).

[8] W. W. Weinstock, "Target Cross Sections Target Models," Part III, Chpt. 3 in Modern Radar, Raymond S. Berkowitz, Ed. (Wiley, New York, 1965).

[9] W. W. Weinstock, "Target Cross Section Models for Radar Systems Analysis," doctoral dissertation, University of Pennsylvania (1964).

[10] F. E. Nathanson, Radar Design Principles (McGraw-Hill, New York, 1969), pp. 148-154.

[11] K. E. Iverson, A Programming Language (Wiley, New York, 1962).

[12] J. D. R. Kramer, Jr., R. M. O'Donnell, and P. F. Gleason, "The Detection Problem for Log-Normally Distributed Signal and Noise," MITRE Technical Report, MTR-136 (May 1972).

[13] D. F. Delong and E. M. Hofstetter, private communication.

[14] F. B. Hildebrand, Introduction to Numerical Analysis (McGraw-Hill, New York, 1959).

[15] P. Swerling, "Recent Development in Target Models for Radar Detection Analysis," AGARD Avionics Tech. Symp. Proc., Istanbul, Turkey, 25-29 May 1970.

[16] R. L. Mitchell and J. F. Walker, "Recursive Methods for Computing Detection Probabilities," IEEE Trans. Aerospace Electron. Syst. AES-7, pp. 671-676 (1971).

[17] I. S. Gradshteyn and I. M. Ryzhik, Tables of Integrals, Series and Products (Academic Press, New York, 1965).

[18] C. Hastings, Jr., Approximations for Digital Computers (Princeton University Press, Princeton, New Jersey, 1955), p. 169.